

Etude, Modélisation et Amélioration des Performances des Convertisseurs Analogique Numérique Entrelacés dans le Temps

Maher Jridi

► **To cite this version:**

Maher Jridi. Etude, Modélisation et Amélioration des Performances des Convertisseurs Analogique Numérique Entrelacés dans le Temps. Micro et nanotechnologies/Microélectronique. Université Sciences et Technologies - Bordeaux I, 2007. Français. tel-00298981

HAL Id: tel-00298981

<https://tel.archives-ouvertes.fr/tel-00298981>

Submitted on 17 Jul 2008

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

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

N° d'ordre : 3517

THÈSE
présentée à
L'UNIVERSITÉ BORDEAUX 1
Ecole doctorale des Sciences Physiques et de l'Ingénieur

par **Maher JRIDI**
POUR OBTENIR LE GRADE DE
DOCTEUR
SPÉCIALITÉ : ÉLECTRONIQUE

ÉTUDE, MODÉLISATION ET AMÉLIORATION DES PERFORMANCES DES
CONVERTISSEURS ANALOGIQUE NUMÉRIQUE ENTRELACÉS DANS LE TEMPS

Soutenue le : 03 décembre 2007

Après avis de :

M. Adel GHAZEL	Professeur	Sup'COM Tunis	Rapporteur
Patrick LOUMEAU	Professeur	ENST Paris	Rapporteur

Devant la commission d'examen formée de :

M. Patrick GARDA	Professeur	LISIF Paris	Président du jury
Dominique DALLET	Professeur	ENSEIRB Bordeaux	Directeur de thèse
Adel GHAZEL	Professeur	Sup'COM Tunis	Rapporteur
Patrick LOUMEAU	Professeur	ENST Paris	Rapporteur
Marc HOURDEQUIN	Docteur Ingénieur	CELAR, Rennes	Industriel
Yannick BERTHOUMIEU	Professeur	ENSEIRB Bordeaux	Examineur
Lilian BOSSUET	Maître de conférences	ENSEIRB Bordeaux	Examineur
Bertrand LE GAL	Maître de conférences	ENSEIRB Bordeaux	Examineur

À ma famille

Remerciements

Ces travaux de recherche ont été effectués au sein du laboratoire d'Intégration du Matériau au Système (IMS) de l'Université Bordeaux 1, dirigé par Monsieur le Professeur Pascal Fouillat. Je tiens à lui exprimer ma profonde reconnaissance pour la confiance qu'il m'a témoignée en m'accueillant au sein du laboratoire IMS.

J'exprime tout particulièrement ma profonde gratitude à Monsieur le Professeur Dominique Dallet qui a assuré la direction de cette thèse, pour la confiance qu'il m'a accordée en m'accueillant au sein de l'équipe "Conception des Circuits et Systèmes Numériques et Mixtes". Son soutien permanent, ses conseils intéressants, ses encouragements et ses aides continus tout au long de ces années de recherche, son dynamisme et sa disponibilité ont joué un rôle déterminant dans ce travail. Quelques mots n'y suffiront pas mais ils témoignent de ma gratitude. Qu'il trouve ici, l'expression de ma grande admiration et de ma profonde reconnaissance.

Au terme de ce travail je tiens à remercier très vivement :

Monsieur Patrick GARDA, Professeur à l'Université Paris VI, pour l'honneur qu'il m'a fait en présidant le jury de soutenance de cette thèse.

Monsieur Adel GHAZEL, Professeur à Sup'Com Tunis, et Monsieur Patrick Loumeau, Professeur à l'ENST Paris, pour l'apport scientifique qu'ils ont ajouté à cette thèse en acceptant d'être rapporteurs. Leurs participations nous honorent.

Messieurs Lilian BOSSUET et Bertrand LE GAL, Maîtres de conférences à l'ENSEIRB Bordeaux, pour leurs participations au jury de cette thèse en tant qu'examinateur, pour leurs efforts relatifs à l'encadrement de ce travail et particulièrement pour leurs sympathies et leurs amitiés.

Messieurs Yannick BERTHOUMIEU, Professeur à l'ENSEIRB Bordeaux, et Marc HOURDEQUIN, Docteur Ingénieur au CELAR/DGA Rennes, d'avoir accepté notre invitation pour participer au jury de thèse.

Monsieur Guy JESTIN du Laboratoire CELAR/DGA, pour l'intérêt qu'il a porté à mes travaux et pour les nombreuses données qu'il m'a fournies.

Monsieur Guillaume MONNERIE, Docteur Ingénieur à NXP , pour toute son aide notamment dans la partie de modélisation mixte du TIADC.

Au cours de mon passage au laboratoire IMS, j'ai eu la chance d'enseigner l'électronique numérique et les télécommunications numériques en tant que tuteur, vacataire et ensuite ATER

à l'Université Bordeaux 1, à l'IUT Bordeaux 1 et à l'ENSEIB. Je tiens tout d'abord à remercier mes étudiants qui m'ont encouragé à continuer ma carrière d'enseignant chercheur et ensuite et avec la plus grande gratitude je tiens à présenter mes vives remerciements à Madame Noëlle LEWIS Maître de conférence à l'Université Bordeaux 1 et responsable du Master 1 EEA, à Monsieur Gyu Phylipe LEGER et à Monsieur Jean THOMAS ,Maîtres de conférences à l'Université Bordeaux 1 pour leurs confiance, leurs soutien et leurs encouragements.

Enfin je souhaite remercier tout le personnel du laboratoire, les informaticiens du laboratoire, tous les doctorants que j'ai pu côtoyer, tous mes amis sans oublier ma famille à la quelle la thèse est dédiée.

MERCI à toutes et à tous.

Table des matières

1	Introduction générale	21
1.1	Contexte: évolution des systèmes de télécommunications mobiles vers la radio logicielle	21
1.2	Problématique: numérisation du signal radio	22
1.3	Contribution: amélioration du facteur de mérite des TIADC	24
1.4	Plan de la thèse	24
2	La conversion analogique numérique	27
2.1	Introduction	27
2.2	Principe de la conversion analogique numérique	28
2.2.1	L'échantillonnage	28
2.2.2	La quantification	30
2.3	Paramètres d'erreurs des CAN	32
2.3.1	Sources de bruit	32
2.3.2	Les erreurs statiques	33
2.3.3	Les erreurs dynamiques	37
2.4	Critères de performance d'un CAN	40
2.4.1	Rapport signal sur bruit	40
2.4.2	Dynamique libre de raie parasite	42
2.4.3	Taux de distortion harmonique	43
2.4.4	Facteur de mérite	44
2.5	Conclusion	44
3	Les convertisseurs analogique numérique entrelacés dans le temps	45
3.1	Introduction	45
3.2	Architectures des CAN	46
3.2.1	Les convertisseurs rapides	47
3.2.2	Les convertisseurs à haute résolution	49

3.2.3	De l'architecture d'un CAN vers les TIADC	51
3.3	Principe des CAN entrelacés dans le temps	52
3.4	Analyse des effets de disparité des erreurs	54
3.4.1	Influence des erreurs statiques	55
3.4.2	Influences des erreurs dynamiques	63
3.4.3	Combinaison des disparités des erreurs	67
3.5	Conclusion	69
4	Étude des méthodes de correction des erreurs de disparités dans les TIADC	71
4.1	Introduction	71
4.2	Classification des méthodes de correction absolues	72
4.2.1	Classification par emplacement de la correction	72
4.2.2	Classification par mode de correction	75
4.2.3	Comparaison des différentes méthodes de correction absolues	76
4.3	Méthodes en différé pour la calibration des TIADC	77
4.3.1	Méthode utilisant des séquences orthogonales	78
4.3.2	Méthode de calibration par FFT	82
4.4	Méthodes en ligne pour la calibration des TIADC	88
4.4.1	Calibration par choix aléatoire de canal	88
4.4.2	Méthode numérique adaptative	93
4.5	Conclusion	98
5	Méthodes de correction proposées	101
5.1	Introduction	101
5.2	Méthode en différé	102
5.2.1	Principe	102
5.2.2	Validations	103
5.2.3	Discussions	105
5.3	Méthode en ligne	106
5.3.1	Principe de la méthode proposée	106
5.3.2	Le filtrage adaptatif	108
5.3.3	Les méthodes d'interpolation	113
5.3.4	Validations	118
5.3.5	Discussions	121
5.4	Amélioration de la solution proposée.	122
5.4.1	Le filtre NLMS	122

5.4.2	Le filtre NDLMS	125
5.5	Conclusion	128
6	Implémentations, tests et mesures	131
6.1	Introduction	131
6.2	La modélisation mixte (analogique-numérique)	132
6.2.1	Environnement de la modélisation mixte	132
6.2.2	La méthodologie de la modélisation mixte	134
6.2.3	Intérêt de la modélisation mixte	136
6.3	Résultats d'implémentation	139
6.3.1	Résultats de synthèse	139
6.3.2	Validation fonctionnelle	143
6.4	Tests et mesures	144
6.4.1	Validation de la méthode en différé avec des données expérimentales . . .	144
6.4.2	Validation de la méthode en ligne par des mesures	146
6.5	Conclusion	150
7	Conclusion générale	153
7.1	Contexte des travaux de la thèse	153
7.2	Contributions et résultats	154
7.3	Analyses critiques	155
7.4	Perspectives	156

Table des figures

1.1	État de l'art des convertisseurs analogique numérique.	23
2.1	Modèle idéal d'un échantillonneur bloqueur.	30
2.2	Caractéristique de transfert idéale d'une loi de quantification uniforme par arrondi.	31
2.3	Erreur d'offset.	33
2.4	Erreur de gain.	34
2.5	Erreur de non-linéarité différentielle.	35
2.6	Évolution de la non-linéarité différentielle.	36
2.7	Évolution de la non-linéarité intégrale.	36
2.8	Erreurs d'ouverture.	37
2.9	Évolution du SNR en fonction de la fréquence du signal d'entrée et du jitter.	39
2.10	Évolution du niveau de bruit moyen de la sortie d'un CAN.	40
2.11	Spectre de sortie d'un CAN de résolution 8 bits.	43
3.1	CAN parallèle par arrondi sur 3 bits.	47
3.2	Convertisseur pipeline.	49
3.3	Convertisseur Sigma Delta.	50
3.4	Convertisseur par approximations successives.	51
3.5	Principe de la conversion par entrelacement.	53
3.6	Modèle mathématique de l'erreur d'offset.	55
3.7	Effet du disparité d'offset sur le spectre d'un TIADC.	57
3.8	Effet du disparité d'offset sur le SNR d'un TIADC.	58
3.9	Modèle mathématique de l'erreur de gain.	58
3.10	Erreur temporelle de la disparité de gain	59
3.11	Effet de la disparité de gain sur le spectre d'un TIADC.	60
3.12	Effet de la disparité de gain sur le SNR d'un TIADC.	61
3.13	Erreur de la <i>NLD</i> pour chaque CAN et pour le TIADC.	62
3.14	Effet des erreurs de non-linéarité pour chaque CAN et pour le TIADC.	64

3.15	Modèle mathématique de l'erreur de retard d'ouverture.	65
3.16	Erreur du retard d'ouverture dans un TIADC.	66
3.17	Effet de la disparité de retard d'ouverture sur le spectre d'un TIADC	67
3.18	Effet de la combinaison des erreurs sur le spectre d'un TIADC.	68
4.1	Le concept de la correction analogique.	73
4.2	Implémentation de la correction analogique.	74
4.3	Le concept de la correction numérique.	74
4.4	Implémentation de la correction numérique.	75
4.5	Illustration de la modulation d'offset.	78
4.6	Synoptique de la calibration d'un TIADC utilisant les séquences orthogonales.	79
4.7	Spectre de sortie du TIADC avant correction.	81
4.8	Spectre de sortie du TIADC après correction.	82
4.9	Procédure de calibration.	84
4.10	Spectre de sortie avant la correction.	86
4.11	Spectre de sortie après la correction.	86
4.12	Évolution du SFDR.	87
4.13	Configuration par choix de deux convertisseurs.	88
4.14	Configuration par choix de trois convertisseurs.	89
4.15	Structure du TIADC utilisant une correction par génération de nombres aléatoires.	90
4.16	Spectre de sortie d'un TIADC à quatre CAN.	91
4.17	Spectre de sortie après correction par choix aléatoire de canal.	91
4.18	Spectre de sortie après correction utilisant la première configuration.	92
4.19	Spectre de sortie après correction utilisant la deuxième configuration.	92
4.20	Calibration du gain pour un seul CAN.	94
4.21	Calibration du gain pour un TIADC à deux CAN.	96
4.22	Calibration d'offset pour un TIADC à deux CAN.	96
4.23	Spectres de sortie avant et après correction.	98
5.1	Principe de la correction pour deux CAN.	102
5.2	Spectres de sortie avant la correction	104
5.3	Spectres de sortie après la correction d'offset	104
5.4	Spectres de sortie après la correction de gain	105
5.5	Évolution du SFDR	106
5.6	Principe de la méthode de correction.	107
5.7	Filtrage optimal de Wiener.	108

5.8	Structure des filtres adaptatifs.	109
5.9	Evolution de l'erreur résiduelle pour les filtres LMS et RLS.	110
5.10	Structure du filtre LMS.	111
5.11	Évolution du temps d'adaptation.	112
5.12	Interpolation au plus proche voisin.	114
5.13	Modèles d'une interpolation linéaire.	114
5.14	Interpolation linéaire.	115
5.15	Interpolation cubique.	116
5.16	Modèle de calcul de l'erreur d'interpolation.	117
5.17	Évaluation de l'erreur d'interpolation.	118
5.18	Sortie temporelle du TIADC.	118
5.19	Évolution de la sortie du filtre FIR.	119
5.20	Évolution de l'erreur résiduelle.	119
5.21	Spectre de sortie avant la correction.	120
5.22	Spectre de sortie après la correction.	120
5.23	Sortie du TIADC avec et sans correction pour des signaux non stationnaires.	121
5.24	Sortie du TIADC avec et sans correction pour un signal dual ton.	122
5.25	Sortie du filtre NLMS pour un signal non stationnaire.	124
5.26	Sortie du filtre NLMS pour un signal en mode dual ton.	125
5.27	Structure du filtre FIR.	125
5.28	Le filtre NDLMS.	126
5.29	Spectres de sortie avant et après correction par filtre NDLMS	128
5.30	Évolution du SNR et du $SFDR$	128
5.31	Spectres de sortie pour un TIADC de résolution 10 bits.	129
5.32	Spectres de sortie pour un TIADC de résolution 12 bits.	129
5.33	Spectres de sortie pour un TIADC de résolution 14 bits.	130
6.1	Méthodologie de conception.	135
6.2	Élément configurable de base des FPGA classiques.	139
6.3	Évolution de la fréquence maximale de fonctionnement après synthèse.	143
6.4	Schéma bloc de test de la méthode de correction implémenté sur FPGA.	143
6.5	Spectre de sortie du TIADC avant et après implémentation de la méthode de correction.	144
6.6	Banc de test.	145
6.7	Spectre de sortie du TIADC avant et après calibration utilisant des données expérimentales.	147

6.8	Banc de mesure pour la méthode en ligne.	148
6.9	Spectre de sortie mesurée à la sortie du TIADC avant implémentation de la méthode de correction.	149
6.10	Spectre de sortie mesurée à la sortie du TIADC après implémentation de la méthode de correction.	149
7.1	Environnement de la co-simulation sous System Generator.	160
7.2	Système d'échantillonnage.	161
7.3	Entrée/sortie du système d'échantillonnage.	161
7.4	Entrée/sortie du système d'échantillonnage modélisé en VHDL-AMS.	163
7.5	Modèle du TIADC.	165
7.6	Histogramme de la distribution gaussienne.	167
7.7	Sorties du quantificateur et du codeur binaire.	170

Liste des tableaux

3.1	Performance d'un CAN.	52
3.2	Paramètres de non-linéarité différentielle de chaque CAN et du TIADC.	63
4.1	Comparaison des méthodes de correction.	77
4.2	Comparaison des méthodes de correction des TIADC.	99
5.1	Comparaison de la complexité des deux filtres LMS et RLS.	110
6.1	Résultats de synthèse avec deux options différentes.	140
6.2	Résultats de synthèse pour différentes génération de FPGA.	142

Notations

δt	erreur de retard d'ouverture
φ	phase du signal d'entrée
ν	pas d'adaptation de l'algorithme adaptatif
ω_0	pulsation du signal d'entrée ($2\pi f_0$)
ω_s	pulsation du signal d'échantillonnage ($2\pi f_s$)
A	amplitude du signal d'entrée
$ERBW$	effective resolution bandwidth
$ENOB$	effective number of bit
f	variable fréquence
f_0	fréquence du signal d'entrée
f_h	fréquence des composantes harmoniques
f_{nh}	fréquence des composantes non-harmoniques
f_s	fréquence d'échantillonnage
G	erreur de gain
M	nombre de convertisseurs entrelacés
MS/s	mega sample per second
NLD	non-linéarité différentielle
NLI	non-linéarité intégrale
N	résolution du convertisseur
O	erreur d'offset
$SINAD$	signal-to-noise and distortion ratio
$SFDR$	spurious-free dynamic range
$SFSR$	signal to full-scale ratio
$SNHR$	signal to non-harmonic ratio
SNR_{th}	theoretical signal-to-noise ratio
SNR	signal-to-noise ratio
t	variable temps
T_0	période du signal d'entrée
T_s	période d'échantillonnage
THD	total harmonic distortion

Acronymes

ASIC	application specific integrated circuit
CAN	convertisseur analogique numérique
CNA	convertisseur numérique analogique
DLMS	delayed least mean squares
DFT	discrete fourier transform : (transformé de fourrier discrète)
DSP	digital signal processor
FIR	finite impulsional response : (réponse impulsionnelle finie)
FFT	fast fourrier transform : (transformé de fourrier rapide)
FPGA	field programmable gate array
IOB	input output block
LMS	least mean squares
LUT	look up table
LVDS	low voltage differential signaling
NLMS	normalized least mean squares
RTL	register transfert level
SOC	system on chip
TIADC	time-interleaved analog to digital converter : (convertisseur analogique numérique entrelacé dans le temps)
VHDL	very high speed integrated circuit hardware description language
VHDL-AMS	very high speed integrated circuit hardware description language analog and mixed signal

Introduction générale

Sommaire

1.1	Contexte: évolution des systèmes de télécommunications mobiles vers la radio logicielle	21
1.2	Problématique: numérisation du signal radio	22
1.3	Contribution: amélioration du facteur de mérite des TIADC	24
1.4	Plan de la thèse	24

1.1 Contexte : évolution des systèmes de télécommunications mobiles vers la radio logicielle

La première génération des systèmes de télécommunications mobiles des années 60 est insuffisante dans le sens où elle ne permet de communiquer que la voie pour des utilisateurs privilégiés en nombre limité. La seconde génération des systèmes cellulaires a connu un succès commercial en donnant la possibilité au grand public de communiquer. L'information transmise est restreinte à la voie et le débit est de 9 kb/s par utilisateur utilisant les bandes de fréquence 890 MHz - 915 MHz et 935 MHz - 960 MHz. Ce débit est revu à la hausse par les extensions du GSM (*Global System Mobile*) au GPRS (*General Packet Radio Service*) et au EDGE (*Enhanced Data Rates for GSM Evolution*). Ensuite, le besoin de communiquer des informations multimédias telles que l'image et la vidéo et le besoin de prendre en charge un nombre croissant d'utilisateurs ont stimulé l'apparition de la troisième génération de la téléphonie mobile, l'UMTS (*Système Uni-*

versel de *Télécommunications Mobiles*). L'UMTS est standardisé en 1999 et occupe des bandes de fréquence différentes de celle du GSM, plus larges et plus importantes : 1920 MHz - 1980 MHz et 2110 MHz - 2170 MHz . Le temps de mise en place de ce réseau qui utilise une infrastructure totalement différente de celle du GSM a imposé une transition progressive du GSM vers l'UMTS. 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, d'un point de vue économique, 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 (*Software Radio*) 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 (voie, 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. La radio logicielle utilise donc les caractères modulaires et programmables de l'informatique et met en œuvre la reconfigurabilité des circuits numériques programmables. Idéalement, l'architecture du récepteur radio est composée de trois éléments : une antenne très large bande, un Convertisseur Analogique Numérique (CAN) ultra rapide et large bande et un système de traitement numérique du signal.

1.2 Problématique : numérisation du signal radio

De nombreuses contraintes technologiques font faces au développement d'architectures pour récepteurs radio logicielle. En particulier, les CAN et les processeurs ne peuvent pas traiter toute la bande du signal radio et encore moins fonctionner aux fréquences radio toute en garantissant une faible consommation. À cela s'ajoutent les problèmes d'intégration d'antennes larges bandes, les contraintes des circuits à faible encombrement, à faible consommation et à coût réduit.

Toutes ces contraintes font que la radio logicielle n'est pas un système à concevoir dans l'immédiat mais plutôt une évolution constante vers un réseau multistandard et multiservice. Aujourd'hui nous parlons de radio logicielle restreinte (*Software Defined Radio*) et des activités de recherche à travers le monde essayent de faire tendre ce système vers la radio logicielle idéale. Ces recherches se focalisent sur trois thématiques : le développement de nouveaux circuits radio fréquence (*front-end analogique*), la proposition de nouvelles architectures de conversion analogique

numérique (considérée comme le principal verrou de la radio logicielle) pour la numérisation du signal radio et l'amélioration des architectures reconfigurables et des algorithmes de traitement numérique des données pour une rapidité accrue.

Dans ce cadre, les travaux présentés dans cette thèse visent l'étude de la conversion analogique numérique afin de mettre en évidence les difficultés à surmonter pour être intégré au plus proche de l'antenne de réception. En effet, le CAN constitue l'interface entre deux traitements différents de l'information et ses performances en termes de rapidité, de précision et de consommation conditionnent celles de toute la chaîne de réception. Dans le cas de la radio logicielle, les CAN reçoivent des signaux radiofréquences larges bandes avec une dynamique très importante. Le nombre de bits nécessaire pour coder le signal est très important et la fréquence d'échantillonnage est très élevée. Ces contraintes de rapidité et de résolution élevées avec une faible consommation figurent aussi dans les applications d'instrumentations et de mesures. Le facteur de mérite des CAN est un paramètre caractéristique de ces circuits englobant la rapidité (en Hertz, Hz), la résolution (en quantum, qu) et la consommation de puissance (en Watt, W). L'évolution de ce paramètre au cours des années est montrée dans la figure 1.1, [1].

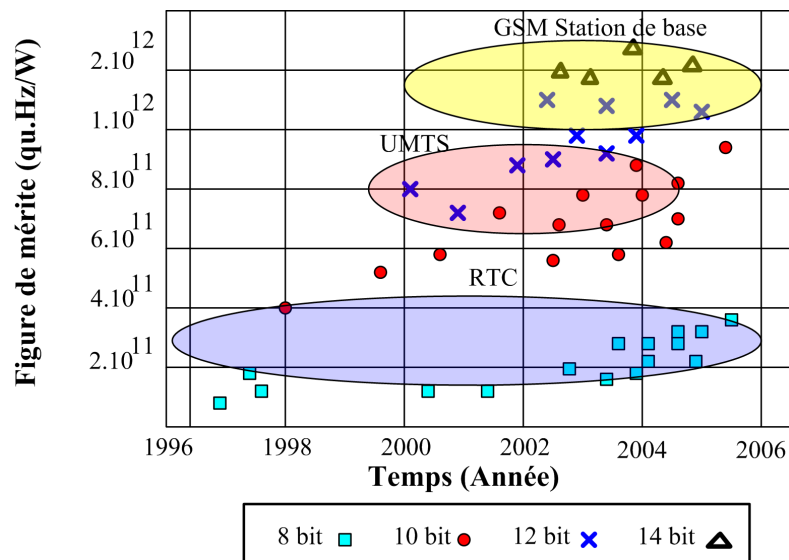


FIG. 1.1 – État de l'art des convertisseurs analogique numérique.

Cette figure montre que pour des applications telles que le RTC (*Réseau Téléphonique Commuté*), l'UMTS ou encore les stations de base des GSM, le choix du CAN est multiple. Le problème est de ne pas avoir un CAN compatible avec le RTC, le GSM et l'UMTS simultanément. En effet pour la radio logicielle, par exemple, le facteur de mérite nécessaire dépasse 6.10^{18} qu.Hz/W pour une consommation comparable à celle du GSM et de l'UMTS. Un CAN seul ne

peut en aucun cas se rapprocher de ces contraintes. Les solutions permettant d'augmenter le taux d'échantillonnage du CAN, sans une augmentation conséquente de la puissance consommée, suscitent de plus en plus d'intérêt. Une d'entre elles est le TIADC pour *Time-Interleaved Analog to Digital Converter*. Il s'agit d'entrelacer dans le temps plusieurs CAN. Le taux d'échantillonnage du système résultant est multiplié par le nombre de CAN utilisés. Malheureusement, les CAN utilisés ne sont pas idéaux, chacun d'entre eux est caractérisé par ses propres erreurs engendrées principalement par les imperfections des processus de fabrication. Les disparités entre les caractéristiques de transfert des différents CAN causent l'apparition de raies spectrales parasites dont les amplitudes et les positions varient en fonction du signal à convertir et du signal d'horloge.

1.3 Contribution : amélioration du facteur de mérite des TIADC

Cette thèse est une contribution à l'étude, à la modélisation et à la correction des erreurs de disparité dans les structures de conversion analogique numérique entrelacées dans le temps. L'objectif que nous avons fixé consiste à évaluer la dégradation des paramètres spectraux suite aux disparités des erreurs statiques telles que l'offset et le gain, et des erreurs dynamiques de retard d'ouverture. Pour se faire, le TIADC est décrit par un modèle comportemental intégrant la partie analogique et numérique dans le même environnement de modélisation et de simulation. Les paramètres spectraux de la sortie du TIADC vérifient les formules théoriques de la perte causée par la disparité des canaux du TIADC. La nécessité d'une méthode de correction est alors mise en évidence. Nous proposons deux méthodes de correction des disparités des erreurs dans un TIADC : la première est une méthode hors ligne nécessitant une mémorisation des données à la sortie du TIADC. La deuxième, en ligne, se base sur l'interpolation et le filtrage adaptatif visant des application de télécommunications.

1.4 Plan de la thèse

Ce mémoire se présente sous la forme de cinq chapitres plus deux chapitres d'introduction et de conclusion :

- Le deuxième chapitre est dédié à l'étude des convertisseurs analogique numérique. Le processus d'échantillonnage et de quantification est alors détaillé. Les caractéristiques des CAN en terme d'erreurs statiques et dynamiques sont illustrées, les paramètres spectraux sont alors déduits.
- Une étude des principales architectures de CAN est présentée dans le troisième chapitre dans le but d'identifier celles qui offrent le meilleur compromis rapidité/résolution. Il a été conclu qu'en terme de rapidité, les convertisseurs flash sont les meilleurs alors que pour la précision les convertisseur de type $\Sigma\Delta$ restent sans concurrent. Dans le souci d'augmenter la vitesse

d'échantillonnage du CAN, les TIADC sont étudiés dans le troisième chapitre. Après avoir détaillé le principe de fonctionnement des ces convertisseurs, une étude sur les répercutions du non appariement des erreurs des différents CAN sur les paramètres spectraux est introduite. Dans un premier temps, les disparités des erreurs d'offset, de gain, de non-linéarité, d'incertitude d'ouverture et du retard d'ouverture sont analysées séparément. Ensuite, l'effet de la combinaison de ces erreurs est mis en exergue afin d'insister sur la nécessité d'une méthode de correction efficace.

- Dans le chapitre 4, une étude non exhaustive mais représentative des méthodes de correction des erreurs dans les TIADC est présentée. Tout d'abord une classification des méthodes de correction sur un seul CAN est établie. Il s'agit des méthodes analogiques ou numériques et des méthodes en lignes et hors lignes. Ensuite, des méthodes développées dans la littérature sont analysées. Leur principe de fonctionnement est détaillé et leur structure est modélisée et vérifiée par des simulations. Enfin, une comparaison quantitative et qualitative est réalisée.
- Afin de pouvoir contribuer à l'amélioration de la résolution effective des TIADC, deux méthodes de correction sont proposées dans le cinquième chapitre. La première est une méthode hors ligne purement mathématique se basant sur le calcul des moyennes pour la détermination des erreurs d'offset dans chaque convertisseur et du rapport des erreurs de gain par rapport à un convertisseur de référence inclus dans la structure entrelacée. La deuxième est une méthode en ligne, qui utilise des notions d'interpolation et de filtrage adaptatif pour rapprocher les caractéristiques de transfert des CAN afin de réduire les disparités entre ces derniers. Des améliorations sont ensuite apportées à cette méthode pour augmenter la rapidité du système et traiter des signaux non stationnaires mais suffisamment lents.
- Le sixième chapitre de ce document détaille les résultats d'implémentations, de tests et de mesures. Dans un premier temps, l'intérêt de la modélisation mixte des TIADC dans un environnement de simulation mixte où le modèle fait interagir des signaux discrets en temps et en amplitude avec des signaux continus est mis en évidence. Pour décrire ce modèle, le choix s'est porté sur le langage VHDL-AMS et le logiciel ADVanceMS. La méthodologie de conception est de type descendante (*TOP-DOWN*). Ensuite, après une brève introduction de la cible matérielle, les résultats de l'implémentation matérielle sont donnés pour évaluer les performances de la solution proposée. Un banc de test fut spécialement mis en œuvre pour tester les deux méthodes de correction proposées. La méthode hors ligne fut testée par des données expérimentales alors que la méthode en ligne fut implémentée dans un FPGA (*Field Programmable Gate Array*) placé derrière deux convertisseurs entrelacés. Les résultats de tests et de mesures sont fournis pour évaluer l'amélioration des performances des TIADC.
- Enfin les conclusions de ce travail sont présentées avec des perspectives à court et à moyen

termes dans le but d'améliorer les solutions proposées et de les intégrer dans un circuit de type ASIC (*Application Specific Integrated Circuit*) avec des outils de routage automatique.

La conversion analogique numérique

Sommaire

2.1	Introduction	27
2.2	Principe de la conversion analogique numérique	28
2.2.1	L'échantillonnage	28
2.2.2	La quantification	30
2.3	Paramètres d'erreurs des CAN	32
2.3.1	Sources de bruit	32
2.3.2	Les erreurs statiques	33
2.3.3	Les erreurs dynamiques	37
2.4	Critères de performance d'un CAN	40
2.4.1	Rapport signal sur bruit	40
2.4.2	Dynamique libre de raie parasite	42
2.4.3	Taux de distortion harmonique	43
2.4.4	Facteur de mérite	44
2.5	Conclusion	44

2.1 Introduction

Les convertisseurs analogique numérique (CAN) constituent l'interface fondamentale entre l'environnement physique, où les signaux sont analogiques, et les circuits de traitement numérique des données, très largement utilisés en raison de leur immunité au bruit, de leur insensibilité au

phénomène de dérive, de leur reconfigurabilité et de la souplesse de leur conception. La grandeur physique (température, pression, lumière, son, image) est convertie par des capteurs en un signal électrique dont les valeurs (tension, intensité) dépendent du phénomène physique mesuré. Les convertisseurs sont devenus à ce titre, un maillon essentiel de l'électronique. Ils sont présents dans la quasi-totalité des circuits mixtes qui contiennent une partie analogique et une partie numérique.

L'opération de conversion analogique numérique, appelée aussi numérisation, se fait en deux étapes distinctes : l'échantillonnage et la quantification. Comme la numérisation s'effectue au moyen de composants électroniques non idéaux, cette opération va engendrer des déformations sur le signal à traiter. Celles-ci seront systématiques par rapport au processus de quantification et aléatoires par rapport à la non idéalité des composants. La première partie de ce chapitre est consacrée à l'étude du principe de fonctionnement du processus d'échantillonnage et de l'opération de quantification. Ensuite, les erreurs dues aux imperfections de l'électronique réalisant les circuits de conversion sont détaillées. Enfin, dans la troisième partie, nous détaillons les critères de performance du CAN. Ceux-ci dépendent des erreurs statiques résultantes de l'espacement non idéal des niveaux de transitions des codes et des erreurs dynamiques induites par des sources d'erreurs supplémentaires.

2.2 Principe de la conversion analogique numérique

L'opération de conversion analogique numérique consiste à transformer un signal continu dans le temps et en amplitude en un signal discrétisé en temps et en amplitude 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. Quelque soit l'architecture du CAN, le processus de conversion passe par deux étapes : l'échantillonnage temporel et la quantification des amplitudes.

2.2.1 L'échantillonnage

L'opération d'échantillonnage est analysée à travers deux niveaux : le niveau système et le niveau circuit.

2.2.1.1 Aspect système

L'échantillonnage consiste à représenter un signal continu dans le temps $s(t)$ par ses valeurs $s(nT_s)$, $n \in \mathbb{Z}$, à des instants multiples de T_s , appelée période d'échantillonnage. L'échantillonnage peut donc être interprété comme la modulation en amplitude d'une distribution $u(t)$ nom-

mée peigne de Dirac par le signal $s(t)$. La représentation mathématique du peigne de Dirac est :

$$u(t) = \sum_{-\infty}^{\infty} \delta(t - nT_s) \quad (2.1)$$

Pour illustrer ce phénomène, nous utilisons un signal sinusoïdal pur $s(t)$, soit :

$$s(t) = A \times \cos(2\pi f_0 t + \varphi) \quad (2.2)$$

où A est l'amplitude maximale, f_0 est la fréquence du signal d'entrée et φ est la phase. Le signal échantillonné, $s_e(nT_s)$, s'écrit alors sous la forme :

$$s_e(nT_s) = A \times \cos(2\pi f_0 nT_s + \varphi) \quad (2.3)$$

D'un point de vue spectrale, l'opération d'échantillonnage affecte le spectre $S_e(f)$ du signal échantillonné. En effet, le spectre du signal échantillonné comprend la fonction $S(f)$, désigné par la bande de base, ainsi que les bandes images qui correspondent à la translation de la bande de base à des multiples entiers de la fréquence d'échantillonnage [2]. Le spectre du signal échantillonné $S_e(f)$ résulte du produit de convolution de $S(f)$ par $U(f)$, $U(f)$ étant le spectre de la fonction peigne de Dirac.

$$S_e(f) = S(f) * U(f) = \sum_{n=-\infty}^{\infty} S\left(f - \left(\frac{n}{T_s}\right)\right) \quad (2.4)$$

Une des caractéristiques fondamentales des signaux échantillonnés est leur périodicité spectrale en raison de la convolution de $S(f)$ par $U(f)$. Restituer le signal d'origine revient donc à supprimer cette périodicité en enlevant les bandes images. Ceci peut être réalisé par un filtre passe bas idéal. Le signal de sortie du filtre $y(t)$ correspond au produit de convolution de la suite $s_e(nT_s)$ par la fonction $h(t)$ qui représente la réponse impulsionnelle de ce filtre idéal, soit :

$$h(t) = \frac{\sin\left(\frac{\pi t}{T_s}\right)}{\left(\frac{\pi t}{T_s}\right)} \quad (2.5)$$

Pour que le signal $y(t)$ soit identique au signal d'origine, il faut que son spectre soit identique à $S(f)$. Ceci n'est possible que si le spectre d'origine ne contient pas de fréquences supérieures à la moitié de la fréquence d'échantillonnage. Dans le cas contraire, la bande image se replie sur la bande de base, d'où le théorème d'échantillonnage de Shannon, qui est satisfait lorsque : $f_s \geq 2f_0$.

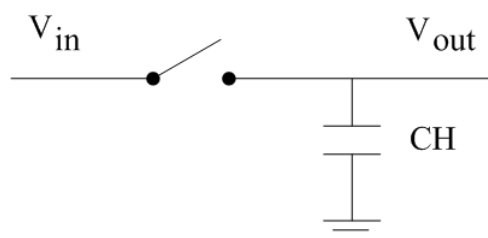


FIG. 2.1 – Modèle idéal d'un échantillonneur bloqueur.

2.2.1.2 Aspect circuit

Dans un convertisseur analogique numérique, l'opération d'échantillonnage est directement suivie par la quantification, qui n'est pas instantanée. Il faut donc maintenir la valeur des échantillons pour assurer la quantification. La manière la plus simple est d'associer un interrupteur à une capacité comme illustré par la figure 2.1 [3]. La capacité joue le rôle d'élément mémoire alors que l'interrupteur réactualise la valeur mémorisée ou l'isole de l'entrée. Dans le cas où l'interrupteur est fermé, l'entrée est transmise sur la sortie : c'est la phase d'échantillonnage. Dans le cas inverse, la sortie reste constante et égale à la dernière valeur du signal transmis : c'est la phase de maintien ou de blocage.

2.2.2 La quantification

2.2.2.1 Principe

La quantification, qui représente la conversion analogique numérique, consiste en l'approximation de chaque valeur du signal échantillonné $s_e(nT_s)$ par un multiple entier d'une quantité élémentaire q appelée pas de quantification ou *LSB* (Least Significant Bit). Toutes les valeurs de sortie du quantificateur sont multiples de cette quantité élémentaire. Pour un q constant, quelque soit l'amplitude du signal d'entrée, la quantification est dite uniforme [4].

L'opération de quantification revient alors à appliquer au signal d'entrée une caractéristique de transfert en marche d'escalier, comme le montre la figure 2.2.

Le centrage de cette caractéristique de transfert définit le type du quantificateur :

- le quantificateur par arrondi, représenté dans la figure 2.2, arrondi à nq toutes les valeurs de l'intervalle $[(n - \frac{1}{2})q, (n + \frac{1}{2})q]$;
- le quantificateur par troncature, qui consiste à approcher par nq toutes valeurs de l'intervalle $[nq, (n + 1)q]$. La caractéristique est donc déplacée de $\frac{q}{2}$ vers la droite sur l'axe des abscisses.

Un des paramètres clés d'un convertisseur analogique numérique est sa résolution N . Le quantum

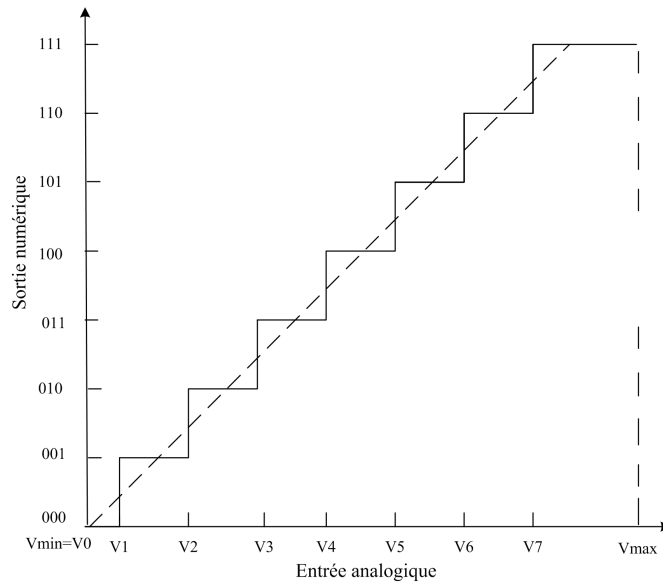


FIG. 2.2 – Caractéristique de transfert idéale d’une loi de quantification uniforme par arrondi.

q dépend de la résolution et de la dynamique d’entrée ($V = V_{max} - V_{min}$) par la relation suivante :

$$q = \frac{V}{2^N} \quad (2.6)$$

2.2.2.2 Erreur de quantification

Si le quantificateur était idéal avec une résolution $N \rightarrow \infty$, la caractéristique de transfert serait une droite sur laquelle une équivalence sera faite entre chaque valeur analogique et le code de sortie. Mais, en réalité, N est de valeur finie. Toute une plage de valeur 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 du CAN exprimé par l’équation 2.2, la sortie du quantificateur idéal s’écrit sous la forme :

$$s(n) = A \times \cos(2\pi f_0 n T_s + \varphi) + e_q(n) \quad (2.7)$$

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 $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]$, $f_s = \frac{1}{T_s}$ dont la valeur efficace est donnée ¹ par :

$$B_q = \frac{q}{\sqrt{12}} \quad (2.8)$$

L'équation 2.8 est vérifiée pour des signaux d'entrées analogiques linéaire (signaux triangulaires par exemple) et pour des signaux sinusoïdaux [2]. Finalement, les équations 2.6 et 2.8 montrent que plus la résolution du CAN est importante, plus la plage analogique définie par le quantum q est réduite, diminuant de fait le bruit de quantification : une meilleure résolution engendre un faible bruit de quantification.

2.3 Paramètres d'erreurs des CAN

L'opération de quantification induit une erreur systématique, mais un CAN est aussi caractérisé par ses paramètres d'erreurs qui sont liés à l'imperfection de l'électronique utilisée et qui font l'objet de cette partie. Avant de lister ces erreurs, les sources physiques des bruits dans les semiconducteurs sont brièvement rappelées afin d'avoir une vue globale sur les bruits existants dans tout circuit électronique.

2.3.1 Sources de bruit

Le bruit dans les circuits et les composants électroniques est dû à des mouvements spontanés et désordonnés qui affectent les charges électriques élémentaires mobiles. Ces mouvements de charge créent des fluctuations de courant et de tension qui imposent une dégradation des propriétés du dispositif considéré [5]. Parmi les sources de bruit, on peut citer trois grandes catégories : les bruits de diffusion, les bruits en excès et les bruits propres aux jonctions [6]. Ces trois familles vont être succinctement énoncées par la suite.

- Bruit de diffusion : le bruit de diffusion dans un composant est lié aux propriétés intrinsèques du substrat. Il est causé par les interactions entre les électrons circulant dans le circuit et le réseau cristallin du semi-conducteur. Le bruit thermique fait partie de cette famille de bruit. Ce bruit est dû aux mouvements aléatoires des porteurs de charges dans un conducteur. Il affecte en particulier les résistances mais également le courant de drain et de grille pour les transistors MOS.
- Bruit en excès : le nom de ce type de bruit vient du fait qu'il s'ajoute au bruit se retrouvant naturellement dans les semi-conducteurs, et qui est prévisible physiquement. Le bruit

¹Le spectre du bruit de quantification s'étend normalement bien au-delà de la fréquence d'échantillonnage. Mais, puisque l'opération de quantification intervient conjointement avec l'échantillonnage, le repliement spectral intervient pour borner la bande de ce bruit.

de Flicker est un de ces bruits. Il s'agit d'un bruit rose ou bruit en $\frac{1}{f}$ en référence à la variation de sa densité spectrale de puissance en $\frac{1}{f}$. Les origines de ce bruit sont difficilement exploitables, mais, il a été reconnu que la raison de la présence de ce bruit est une variation de la conductivité du matériau. Il affecte les résistances, les transistors MOS et le courant de collecteur/émetteur des transistors bipolaires.

- Bruit de jonction : il s'agit ici de la jonction des semiconducteurs. Le bruit de jonction le plus connu est la bruit de grenaille. Son origine est liée à la traversée d'une barrière de potentiel par des porteurs de charge. Il s'agit d'un bruit blanc qui affecte principalement les diodes et les transistors bipolaires.

Ces sources physiques de bruit au niveau des semiconducteurs induisent des erreurs statiques et dynamiques dans les CAN.

2.3.2 Les erreurs statiques

La description statique du quantificateur idéal présenté dans la première partie 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 existe définit la nature de l'erreur que nous exposons dans les paragraphes suivants, [7].

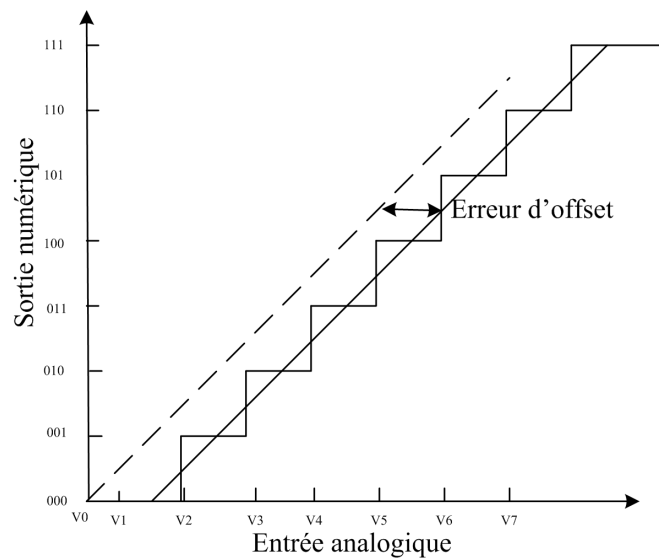


FIG. 2.3 – Erreur d'offset.

2.3.2.1 Erreur d'offset

L'erreur d'offset, comme le montre la figure 2.3, est un décalage en tension introduit par le convertisseur sur l'ensemble du signal. Connue aussi sous le nom d'erreur sur le zéro analogique,

cette erreur est exprimée en *LSB* ou en pourcentage de la pleine échelle. Il s'agit d'une constante $O \in \mathbb{Z}$ additive au signal de sortie du CAN et dans le cas idéal, cette erreur vaut zéro. Comparée à l'équation 2.7, la sortie du CAN en présence d'une erreur d'offset, aura la forme suivante :

$$s(n) = O + A \cos(2\pi f_0 n T_s + \varphi) + e_q(n) \quad (2.9)$$

où $e_q(n)$ désigne le signal d'erreur de quantification.

2.3.2.2 Erreur de gain

L'erreur de gain consiste en un changement de la pente de la caractéristique de transfert idéale conformément à la figure 2.4. Cette erreur peut altérer la pleine échelle du convertisseur et par la suite la valeur du quantum. Il s'agit d'une constante $G \in \mathbb{Z}$ multiplicative du signal de sortie et dans le cas idéal, cette erreur vaut 1. Elle est exprimée en *LSB* ou en pourcentage de la pleine échelle. Comparée à l'équation 2.7, la sortie du CAN en présence d'une erreur de gain, aura la forme suivante :

$$s(n) = G \times A \cos(2\pi f_0 n T_s + \varphi) + e_q(n) \quad (2.10)$$

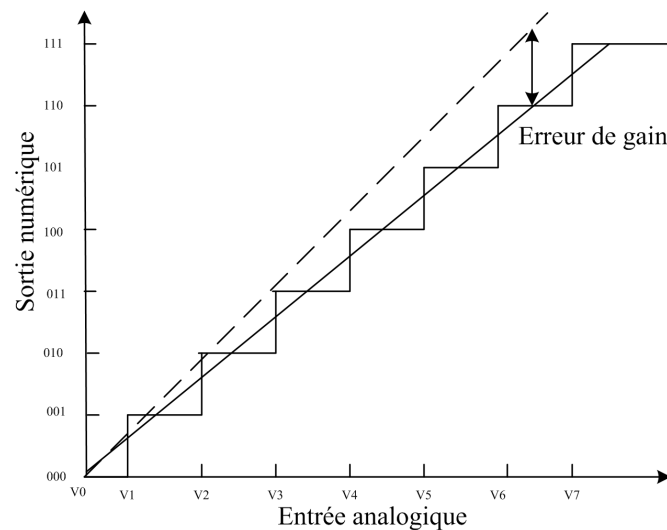


FIG. 2.4 – Erreur de gain.

2.3.2.3 Erreur de non-linéarité

Les erreurs de non-linéarités reflètent des variations locales, ne pouvant pas s'exprimer de façon linéaire, des seuils analogiques de transition de la caractéristique de transfert. Deux paramètres

de non linéarité sont définis : la *NLD* (non-linéarité différentielle) et la *NLI* (non-linéarité intégrale). Ces paramètres ne sont déterminés une fois les erreurs d'offset et de gain corrigées.

- **NLD** : Pour un code i , la $NLD(i)$ correspond à l'écart entre la largeur réelle de la marche associée à ce code, q_i et la largeur théorique, q de la caractéristique de transfert. Cette valeur est normalisée par q et est exprimée en *LSB* [8].

$$NLD(i) = \frac{q_i - q}{q} \quad (2.11)$$

Dans la littérature, le terme *NLD* d'un convertisseur définit le maximum de toutes les $NLD(i)$ calculées sur tous les codes de sortie du CAN, figure 2.5.

$$NLD = \max(NLD(i)) \quad (2.12)$$

Dans le cas idéal, cette valeur vaut zéro. Si cette valeur excède -1 LSB, le convertisseur présente des codes manquants.

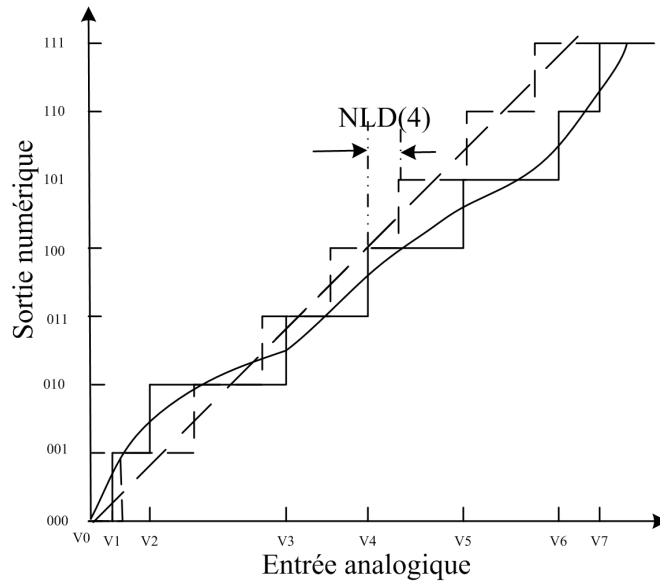


FIG. 2.5 – Erreur de non-linéarité différentielle.

- **NLI** : La NLI du code i , exprime la somme cumulée des *NLD* jusqu'au rang i :

$$NLI(i) = \sum_{j \leq i} NLD(j) \quad (2.13)$$

L'erreur de linéarité intégrale du code i représente l'écart entre le centre réel du palier i et son centre idéal théorique. Réciproquement, la courbe de transfert réelle peut être déduite de la droite de transfert idéale en y ajoutant la *NLI* code par code.

Les paramètres de non-linéarité sont calculés par une technique utilisant l'analyse statistique se basant sur le calcul des histogrammes. La figure 2.6 donne l'allure de l'erreur de non-linéarité différentielle d'un CAN de résolution 8 bits. L'analyse par histogramme permet de déterminer les codes manquants correspondant à une fréquence d'apparition nulle, donc à une valeur de NLD égale à -1 LSB. La méthode d'analyse par histogramme permet d'extraire la valeur de la NLI dont l'allure est montrée dans la figure 2.7.

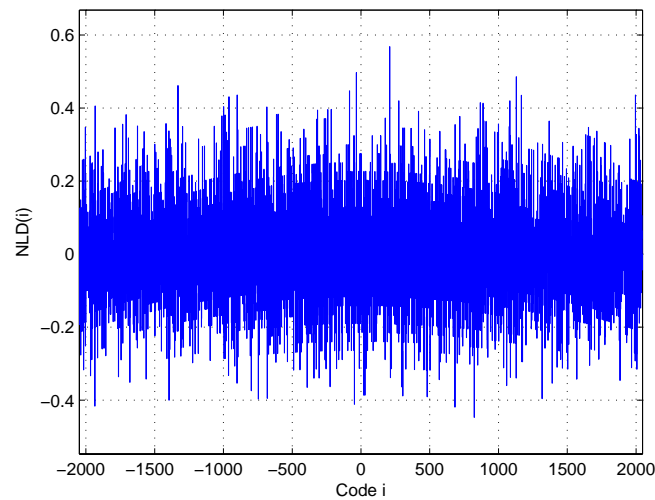


FIG. 2.6 – Évolution de la non-linéarité différentielle.

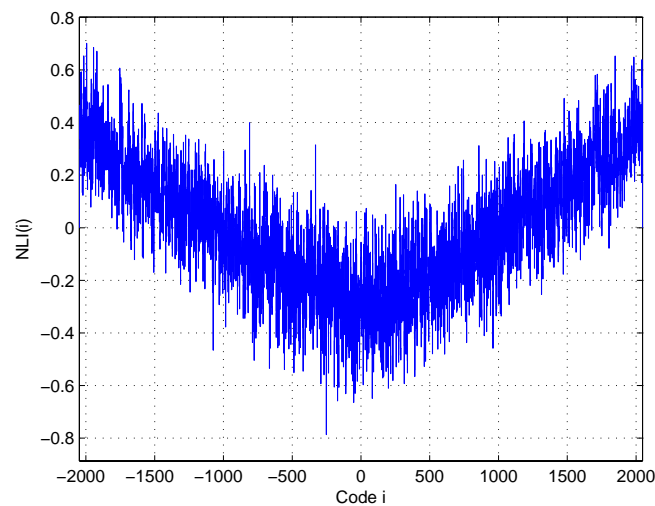


FIG. 2.7 – Évolution de la non-linéarité intégrale.

2.3.3 Les erreurs dynamiques

2.3.3.1 Types d'erreurs temporelles

Contrairement aux erreurs statiques résultant de l'espacement non idéal des niveaux de transition des codes, les erreurs dynamiques sont résultantes de sources supplémentaires d'erreurs (du CAN et/ou de son environnement) induites par la variation du signal analogique.

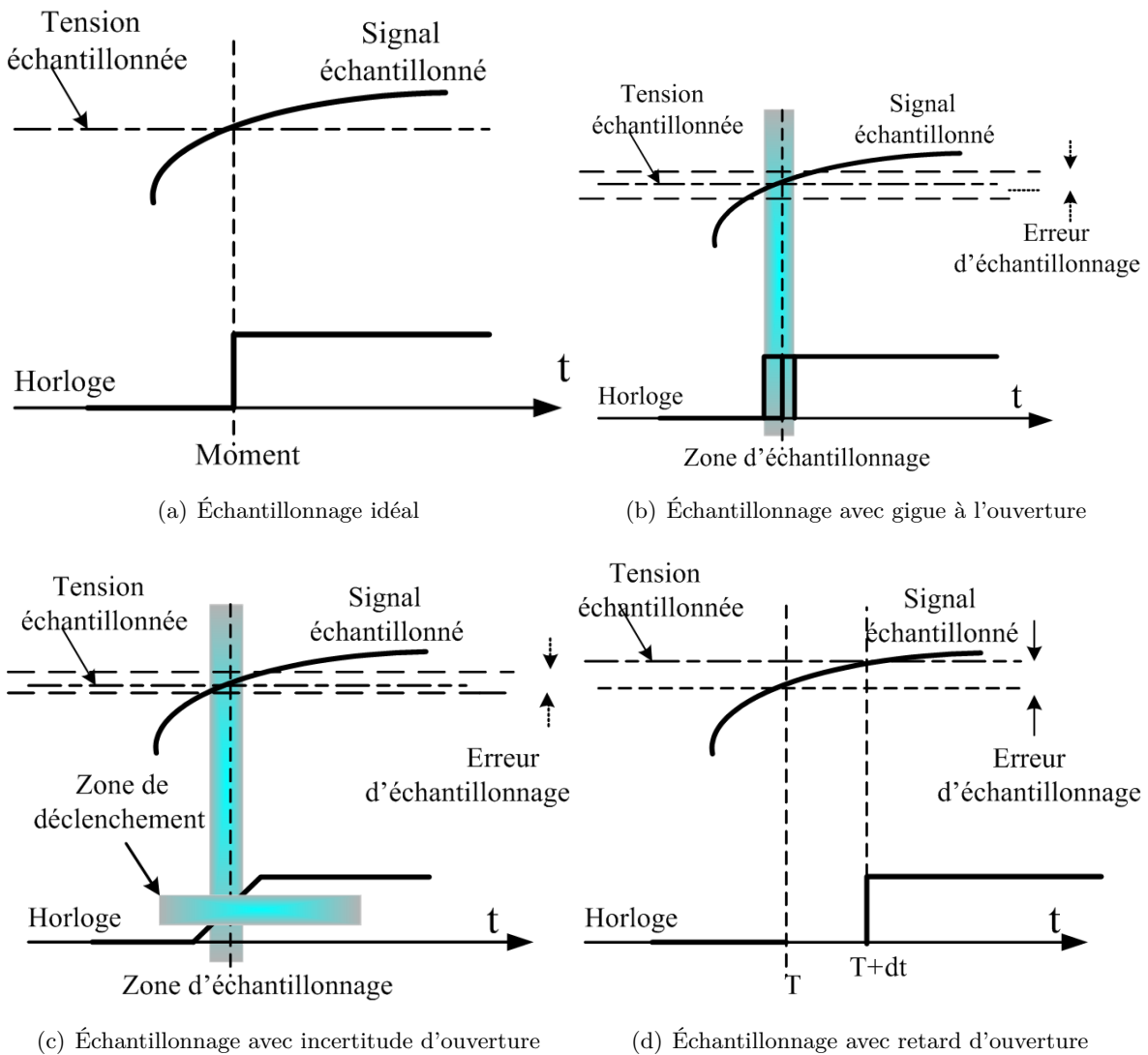


FIG. 2.8 – Erreurs d'ouverture.

Les trois principaux types d'erreurs dynamiques sont liés à l'échantillonnage et sont la gigue à l'ouverture (*aperture jitter*), l'incertitude à l'ouverture (*aperture uncertainty*) et le retard à l'ouverture (*aperture delay* ou *aperture time*) [9].

– **Gigue à l'ouverture :**

C'est l'une des plus importantes sources d'erreurs sur les convertisseurs car une erreur sur l'instant de capture entraîne une erreur sur la tension échantillonnée. La gigue à l'ouverture est due à des variations aléatoires de l'instant d'échantillonnage causées par les sources de bruit (bruit thermique, bruit d'alimentation, bruit d'horloge, etc).

– **Incertitude à l'ouverture :**

Il ne faut pas confondre la gigue à l'ouverture qui est due à une variation temporelle de la position du front de l'horloge avec l'incertitude à l'ouverture qui est due à une variation du potentiel de seuil pour lequel l'échantillonnage a lieu. Contrairement à la gigue à l'ouverture, l'incertitude à l'ouverture diminue avec l'augmentation de la pente des fronts d'horloge : si la pente est infinie, l'incertitude à l'ouverture devient nulle. Mais pour obtenir un front d'horloge raide, il faut concevoir un circuit dans lequel la consommation augmente.

– **Retard à l'ouverture :**

Le retard à l'ouverture d'un convertisseur caractérise le temps entre l'instant où il reçoit l'ordre d'échantillonner et le moment où il le fait. Ce retard est essentiellement dû au temps de transition dans les composants, au temps de propagation dans les lignes métalliques et au temps de commutation de l'horloge.

Contrairement aux erreurs précédentes qui créent une erreur de nature aléatoire, le retard à l'ouverture est fixe. Il est de loin le plus important mais le fait qu'il soit fixe le rend beaucoup moins problématique, car il peut être pris en compte ou compensé.

La figure 2.8 montre les différents impacts de ces erreurs sur le signal échantillonné.

2.3.3.2 Influence de l'erreur du jitter

L'erreur de jitter est la traduction dans le domaine temporel du bruit de phase qui est un phénomène due à l'accumulation de plusieurs sources de bruit dans le semi-conducteur.

Par sa définition même, le jitter affecte le processus d'échantillonnage. Considérons un système d'échantillonnage cadencé à $t = nT_s + \varepsilon$, ε étant une variable aléatoire qui modélise le jitter et qui suit une distribution Gaussienne $N(0, \sigma^2)$. L'entrée de ce système est un signal analogique $s(t) = A \cos(2\pi f_0 t)$. La sortie du système d'échantillonnage dans le cas idéal est :

$$s(n) = A \cos(2\pi f_0 n T_s + \varphi) \quad (2.14)$$

Mais en présence de jitter, la sortie devient :

$$s_e(n) = A \cos(2\pi f_0 (n T_s + \varepsilon) + \varphi) \quad (2.15)$$

L'erreur introduite par le jitter est donc obtenue en faisant la différence entre les deux dernières équations. En supposant que $2\pi f_0 \varepsilon \ll 1$ et avec un développement limité du premier ordre des

fonctions cosinus et sinus, on en déduit que l'erreur induite par le bruit de jitter est :

$$\begin{aligned} \text{erreur}_j = s_e(n) - s(n) &= (A \cos(2\pi f_0 n T_s + \varphi) \times \cos(2\pi f_0 \varepsilon)) \\ &- (A \sin(2\pi f_0 n T_s + \varphi) \times \sin(2\pi f_0 \varepsilon)) - A \cos(2\pi f_0 n T_s + \varphi) \end{aligned} \quad (2.16)$$

qui peut s'écrire sous la forme :

$$\text{erreur}_j \approx (-A 2\pi f_0 \varepsilon) \sin(2\pi f_0 n T_s + \varphi) = \varepsilon \left(\frac{\partial s(t)}{\partial t} \right)_{t=nT_s} \quad (2.17)$$

Naturellement, cette erreur dépend du jitter ε mais aussi de $\frac{\partial s(t)}{\partial t}$ pour des signaux sinusoïdaux à l'entrée du CAN. Par conséquent, l'effet du jitter est d'autant plus important que la fréquence d'entrée f_0 augmente. C'est le cas d'un système large bande par exemple. La puissance du bruit dû au jitter est de la forme :

$$P_j = 2\pi^2 f_0^2 A^2 \sigma^2 \quad (2.18)$$

Ainsi, en présence de l'erreur du jitter, le rapport signal sur bruit (SNR , détaillé dans la section suivante) résultant est :

$$SNR = 10 \log_{10} \frac{\frac{A^2}{2}}{2\pi^2 f_0^2 A^2 \sigma^2} = -20 \log_{10} (2\pi f_0 \sigma) \quad (2.19)$$

La figure 2.9 montre que ce paramètre diminue en augmentant la fréquence du signal d'entrée pour différentes valeurs de jitter. La dégradation du SNR par rapport à l'augmentation de la

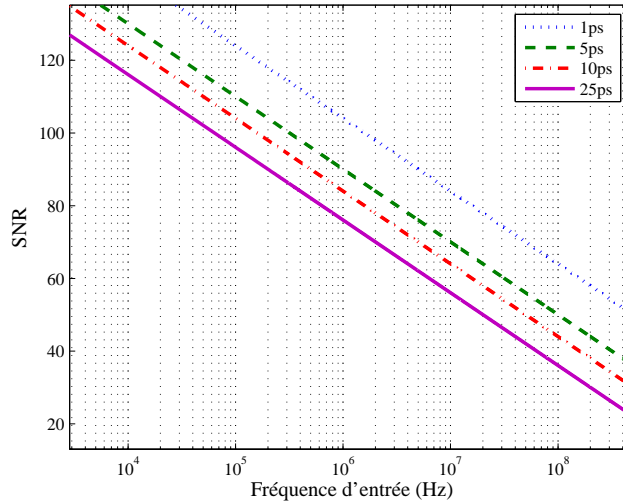


FIG. 2.9 – Évolution du SNR en fonction de la fréquence du signal d'entrée et du jitter.

valeur du jitter est due à l'élévation du niveau du bruit moyen comme le montre les spectres de la figure 2.10. Ces résultats de simulation sont réalisés pour un modèle de CAN de résolution 8 bits et de fréquence d'échantillonnage égale à 500 MS/s.

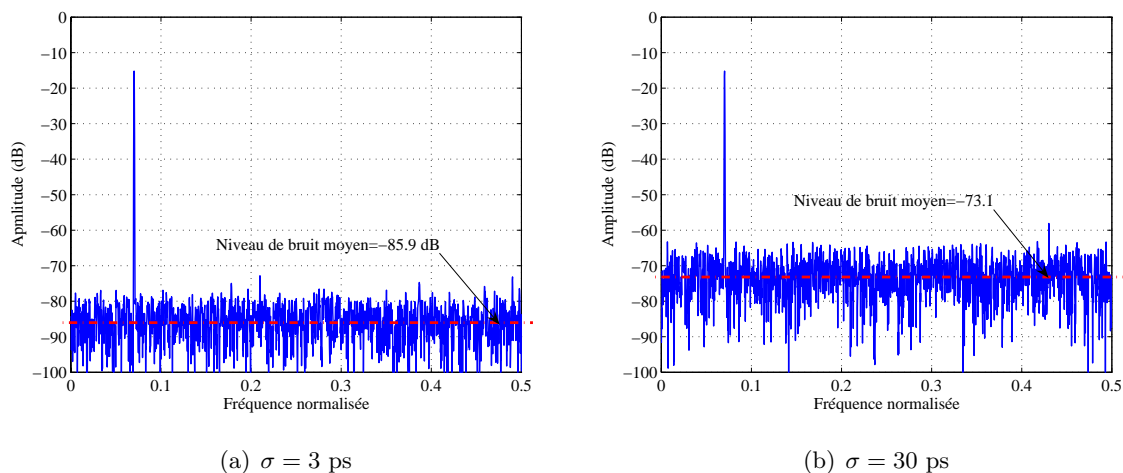


FIG. 2.10 – Évolution du niveau de bruit moyen de la sortie d'un CAN.

2.4 Critères de performance d'un CAN

Du fait qu'il existe toute une gamme de CAN pour différentes applications, un ensemble de critères est défini pour évaluer les performances des convertisseurs. Cependant, les critères de choix d'un CAN 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 CAN 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 le CAN peut altérer le fonctionnement globale du convertisseur [10]. Par ailleurs, nous définissons les paramètres de performance en tant qu'ensemble de variables permettant d'évaluer les CAN. Dans la suite de cette section, les différents paramètres caractérisant les CAN sont définis.

2.4.1 Rapport signal sur bruit

Le rapport signal sur bruit SNR (*Signal to Noise Ratio*) et le rapport signal sur bruit plus distortions harmonique $SINAD$ (*Signal to Noise And Distortion Ratio*) sont deux des paramètres les plus utilisés pour l'évaluation des performances d'un convertisseur. La différence entre ces deux paramètres consiste dans la définition du bruit. En effet, toutes variations entre le signal

de sortie du convertisseur et son entrée sans prendre en compte :

1. des harmoniques de la composante fondamentale,
2. de la composante DC,
3. des déviations causées par la réponse non linéaire du système,

sont attribuées au bruit. Le *SINAD* est défini comme étant le rapport de la valeur efficace du signal par la valeur efficace du bruit plus toutes les distortions. Le rapport signal sur bruit ou *SNR* est défini comme étant le rapport entre la puissance du signal et la puissance du bruit sans les harmoniques. Pour bien définir le bruit en question, le standard IEEE 1241 [11] définit ce terme par le *SNHR* (*Signal to Non-Harmonic Ratio*). Le *SNR* ou le *SNHR* dépend de l'amplitude du signal d'entrée. Ces valeurs doivent être spécifiées dans la description du composant. Pour un convertisseur idéal, le seul bruit étant celui lié au processus de quantification dont la valeur efficace est donnée dans l'équation (2.8).

Dans le cas d'un CAN par arrondi, l'erreur de quantification est comprise dans l'intervalle $[-0.5q, 0.5q]$ autour d'une valeur moyenne nulle pour un signal d'entrée de forme sinusoïdale d'amplitude crête à crête égale à $PE - q$, PE représente la pleine échelle de la conversion. La valeur efficace de ce bruit est :

$$B_q = \frac{1}{\sqrt{12}} \quad (2.20)$$

Il est alors possible de déterminer le SNR_{th} théorique d'un CAN idéal de résolution N . La valeur efficace d'un signal sinusoïdal dont l'amplitude est égale à la pleine échelle du CAN est donné par :

$$V_{eff} = \frac{2^N}{2\sqrt{2}} \quad (2.21)$$

Le rapport des équations 2.21 par 2.20 exprimé en décibel (dB) définit le SNR_{th} théorique du CAN :

$$SNR_{th} = 20 \log \left(\frac{V_{eff}}{B_q} \right) = 20 \log \left(\frac{2^N \sqrt{12}}{2\sqrt{2}} \right) = 6.02N + 1,76 \quad (dB) \quad (2.22)$$

Le fait de considérer particulièrement un signal sinusoïdal est due à son utilisation dans le test des convertisseurs. Néanmoins, l'équation 2.22 reste valable pour des signaux triangulaires d'amplitude crête à crête égale à $PE - q$. Pour des signaux de type Gaussien avec un écart type $\sigma = \frac{PE}{4}$ le SNR_{th} théorique devient, [12] :

$$SNR_{th} = 20 \log \left(\frac{V_{eff}}{B_q} \right) = 6.02N - 7.3 \quad (dB) \quad (2.23)$$

Mais, souvent, l'amplitude crête à crête du signal à convertir ne couvre pas exactement la pleine échelle PE . Dans ces conditions, il faut tenir compte du paramètre exprimant le rapport de

l'amplitude du signal par la pleine échelle du CAN ($SFSR$, *Signal to Full Scale Ratio*) pour le calcul du SNR_{th} . Le $SFSR$ est exprimé en dB par la relation 2.24 :

$$SFSR = 20 \log \frac{2A}{PE} \quad (dB) \quad (2.24)$$

La valeur du SNR_{th} dans ces conditions, pour un signal sinusoïdal, est :

$$SNR_{th} = 6.02N + 1.76 + SFSR \quad (dB) \quad (2.25)$$

Cette valeur du SNR_{th} théorique sert de référence dans l'évaluation des paramètres SNR et $SINAD$ d'un convertisseur réel. Dans cette section nous avons exprimé la valeur du SNR_{th} en ne tenant compte que de l'erreur de quantification. Dans un CAN réel, le bruit apporté par toutes les imperfections du circuit diminue la valeur du SNR_{th} . Après avoir mesuré le SNR d'un CAN pour un signal d'entrée sinusoïdal dont l'amplitude est égale à la pleine échelle, le nombre de bits effectif ($ENOB$, *Effectif Number Of Bit*) est déduit à partir de la relation 2.26 :

$$ENOB = \frac{SNR - 1.76}{6.02} \quad (2.26)$$

Enfin, un autre paramètre peut être extrait après le calcul du SNR par analyse fréquentielle du CAN. Pour une fréquence d'échantillonnage fixe, la valeur du SNR est constante pour les faibles fréquences du signal d'entrée (par rapport à f_s) et décroît lorsque la fréquence augmente. Ce fait s'explique par l'augmentation de l'influence du jitter pour les hautes fréquences du signal d'entrée. La valeur de la fréquence f pour laquelle le SNR diminue de 3 dB se nomme bande passante du convertisseur ($ERBW$, *Effective Resolution BandWith*). Ce paramètre indique la bande de fréquence dans laquelle le convertisseur doit être utilisé.

2.4.2 Dynamique libre de raie parasite

Les signaux de sortie des CAN sont généralement caractérisés par la combinaison des composantes harmoniques situées à des fréquences multiples de la fréquence du signal d'entrée et de persistantes composantes non-harmoniques. Ces deux types de composantes dégradent les performances du convertisseur. Le $SFDR$ (*Spurious Free Dynamic Range*) est défini comme le rapport entre l'amplitude du spectre de sortie du convertisseur à la fréquence d'entrée f_0 et l'amplitude de la plus grande valeur des composantes harmoniques et/ou non-harmoniques observées sur la bande de Nyquist. La formule suivante exprime la valeur du $SFDR$ en dB :

$$SFDR = 20 \log \left\{ \frac{|X(f_0)|}{\max_{f_{nh}, f_h} \{|X(f_{nh})| \text{ ou } |X(f_h)|\}} \right\} \quad (2.27)$$

où :

- $|X|$ est le module du spectre de la sortie du convertisseur,
- f_h les fréquences des composantes harmoniques,
- f_{nh} les fréquences des composantes non-harmoniques.

La définition littérale du *SFDR* décrite dans cette section est illustrée par la figure 2.11. Celle-ci est représentative d'une simulation d'un modèle de CAN de résolution 8 bits avec insertion d'erreurs statiques qui a provoqué l'apparition de raies harmoniques.

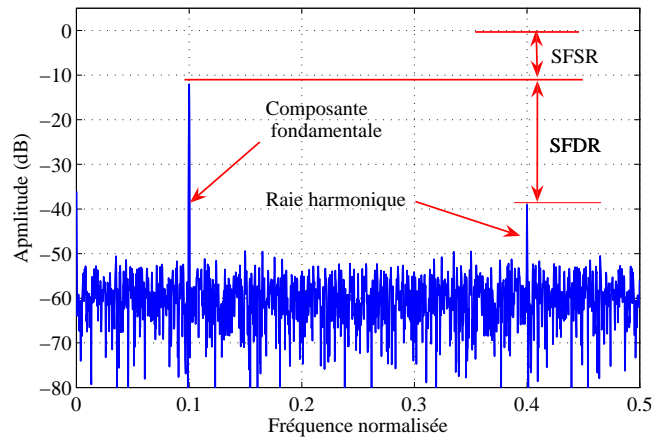


FIG. 2.11 – Spectre de sortie d'un CAN de résolution 8 bits.

2.4.3 Taux de distortion harmonique

Le Taux de distortion harmonique *THD* (*Total Harmonic Distortion*) est un paramètre de mesure de la distortion harmonique causée par la non-linéarité du convertisseur. Il s'exprime par le rapport entre la somme des valeurs efficaces des composantes harmoniques situées aux fréquences $k.f_0$, et la valeur efficace du signal d'entrée :

$$THD = 10 \log \sqrt{\frac{\sum_h (X(f_h))^2}{(X(f_0))^2}} \quad (2.28)$$

où $X(f_h)$ est l'amplitude de la h^{eme} composante harmonique du spectre de la sortie du convertisseur. Pour un CAN idéal, la valeur du *THD* tend vers l'infini, mais les harmoniques dues aux imperfections du CAN réel font que cette valeur se dégrade. Généralement, on se restreint aux 5 premières harmoniques qui sont les plus significatifs.

2.4.4 Facteur de mérite

Afin de comparer deux convertisseurs, un critère plus général que les paramètres précédemment cités est adopté. Il s'agit du facteur de mérite *FOM* (*Figure Of Merit*) défini dans [13] par :

$$FOM = \frac{2^{ENOB} 2.ERBW}{P_m} \quad (2.29)$$

P_m est la puissance moyenne totale consommée par le CAN et *ENOB* est calculé par l'équation 2.26 pour une fréquence d'entrée faible.

D'autres définitions moins utilisées font intervenir l'espace S en m^2 du circuit conformément à l'équation 2.30 [4] et [14] :

$$FOM = \frac{2^{ENOB} 2.ERBW}{S.P_m} \quad (2.30)$$

Dans les récentes publications scientifiques, les différents paramètres de calcul du *FOM* ne sont pas donnés. Le plus souvent, il manque le *ERBW*. Les récents états de l'art des CAN fixent la bande passante effective à la fréquence de Nyquist et le calcul de *ENOB* se fait pour des faibles fréquences du signal d'entrée. Il en résulte des estimations optimistes de la valeur du *FOM*.

2.5 Conclusion

Dans ce chapitre nous avons rappelé le principe de la conversion analogique numérique avec un regard particulier sur les paramètres d'erreurs des CAN et leurs influences sur les performances des convertisseurs analogique numérique.

Tout d'abord, nous avons mis l'accent sur l'erreur systématique de la quantification et sur son influence sur le *SNR*. A cette erreur intrinsèque s'ajoutent les effets des imperfections du composant. Nous avons listé dans la deuxième partie de ce chapitre l'ensemble des erreurs statiques (offset, gain et non-linéarité) et dynamiques (jitter et retard à l'ouverture) caractérisant un CAN.

Ces erreurs créent des raies parasites harmoniques et non-harmoniques au niveau du spectre du signal converti et dégradent les paramètres spectraux des CAN (*SNR*, *SFDR*, *THD*,...) ainsi que le facteur de mérite qui est un critère utilisé pour l'évaluation des performances des CAN. Pour les systèmes de communications très large bande et pour la radio logicielle, la valeur du *FOM* n'est pas suffisante (la quasi majorité des CAN ont des *FOM* compris entre 10^{11} et 10^{12} bit.Hz/W). Cette valeur peut être augmentée par l'utilisation de technologies plus avancées, mais cela se fait toujours au dépend du coût. Le chapitre suivant présente une nouvelle technique pour augmenter la fréquence d'échantillonnage du CAN, et donc du *FOM*, sans augmentation conséquente de la consommation ni du coût.

Les convertisseurs analogique numérique entrelacés dans le temps

Sommaire

3.1	Introduction	45
3.2	Architectures des CAN	46
3.2.1	Les convertisseurs rapides	47
3.2.2	Les convertisseurs à haute résolution	49
3.2.3	De l'architecture d'un CAN vers les TIADC	51
3.3	Principe des CAN entrelacés dans le temps	52
3.4	Analyse des effets de disparité des erreurs	54
3.4.1	Influence des erreurs statiques	55
3.4.2	Influences des erreurs dynamiques	63
3.4.3	Combinaison des disparités des erreurs	67
3.5	Conclusion	69

3.1 Introduction

La vitesse et la précision des CAN sont deux paramètres qui dépendent essentiellement de la technologie utilisée et de l'architecture du convertisseur. Nous avons décrit dans le chapitre précédent les paramètres d'erreurs essentiels pour la description comportementale d'un CAN. Ces différents paramètres donnent une évaluation de la dégradation du signal numérisé. Dans ce chapitre une liste non exhaustive de certaines architectures de CAN est présentée. Ces architectures sont limitées en fréquence d'échantillonnage et/ou précision et ce constat devient critique

pour certaines applications nécessitant vitesse et précision (radio logicielle, mesures,...). Une solution pour multiplier la fréquence d'échantillonnage d'un convertisseur consiste à entrelacer plusieurs CAN dans le temps : TIADC pour *Time Interleaved ADC*. Le principe de ce convertisseur est détaillé dans la deuxième partie du chapitre. Les facteurs limitant la résolution de ces convertisseurs sont liés aux défauts technologiques aléatoires. En effet, les erreurs statiques des différents CAN utilisés dans la structure entrelacée ne sont pas forcément rigoureusement identiques, causant ainsi des problèmes d'appariement. L'origine des problèmes d'appariement entre les CAN est principalement liée aux défauts de fabrication ainsi qu'aux effets du vieillissement. La troisième partie du chapitre est donc consacrée à l'étude des effets individuels du non appariement des différentes erreurs caractérisant un TIADC. La combinaison de toutes les erreurs est analysée en fin de chapitre.

3.2 Architectures des CAN

Dans cette partie nous présentons les architectures de base utilisées actuellement pour réaliser la conversion analogique numérique. Si l'on fait abstraction des convertisseurs à rampe (ou à intégration) quelque peu tombés en désuétude¹, du fait de leur lenteur souvent rédhibitoire et de leur faible bande passante, quatre grandes familles de CAN se partagent le marché depuis de longues années. D'un côté les convertisseurs précis à approximations successives (ou SAR, *Successive Approximation Register*) et SigmaDelta (ou Delta-Sigma). De l'autre, les convertisseurs rapide parallèles et pipeline.

Pour des débits en sortie de CAN de 10 MS/s (Mega Samples/s) et plus, les deux premières familles citées ne sont pas utilisées, bien qu'elles se rapprochent de cette barre symbolique. Par exemple, dans la catégorie des CAN de résolution 16 bits, les convertisseurs sigma-delta et SAR atteignent aujourd'hui un débit de 5 MS/s (ADS1605 de Texas Instrument) et de 3 MS/s (AD7621 d'Analog Devices) respectivement. Quant aux convertisseurs parallèles et pipeline, à défaut de révolution, de pertinentes innovations leur permettent de repousser sans cesse leurs limites en termes de fréquence d'échantillonnage et de résolution, mais malheureusement souvent au détriment de la consommation.

¹Ces convertisseurs sont toutefois encore en usage dans certains systèmes (multimètres numériques par exemple) grâce à leur précision et à une excellente réjection du bruit.

3.2.1 Les convertisseurs rapides

3.2.1.1 Les convertisseurs parallèles

Dès lors qu'il est question de conversion rapide, le CAN parallèle (ou *flash*) reste sans rival [15]. Il met en oeuvre une architecture parallèle dans laquelle le signal à convertir sur N bits est simultanément comparé à de multiples tensions de référence comme le montre la figure 3.1. Les

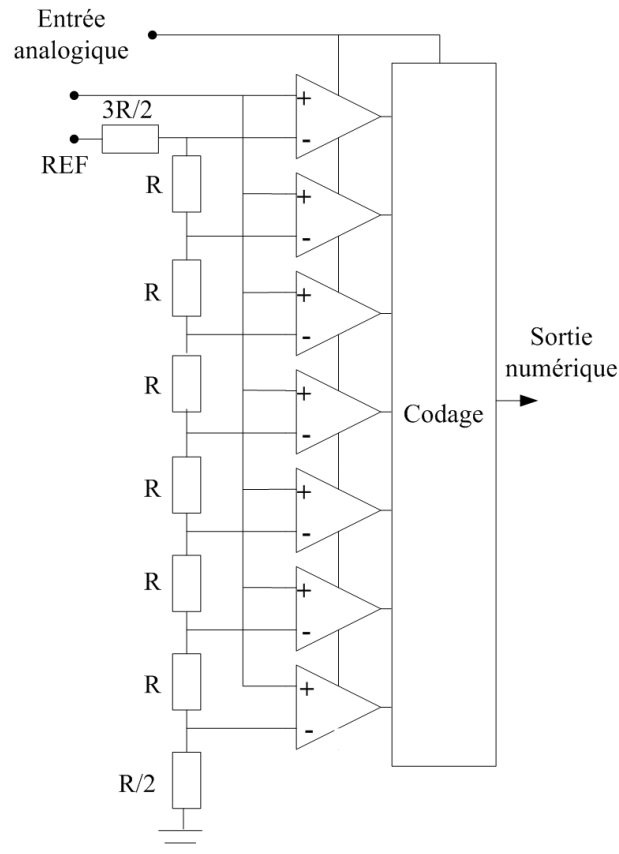


FIG. 3.1 – CAN parallèle par arrondi sur 3 bits.

valeurs des tensions de référence sont espacées de 1 LSB (ou un quantum) et fixées par un réseau de résistances. Au fur et à mesure que la tension croît, les sorties des comparateurs basculent de l'état bas à l'état haut. Ce mode de fonctionnement s'apparente à celui d'un thermomètre à mercure lorsque la température augmente. De là est issu le nom de 'code thermomètre' parfois donné au code généré en sortie.

Puisque les sorties des comparateurs ne sont pas directement exploitables, une logique de transcodage se charge alors de délivrer les données finales selon un format binaire conventionnel sur N bits. L'inconvénient majeur du convertisseur parallèle est le nombre de comparateurs qui croît de façon exponentielle avec la résolution. L'investissement matériel se révèle alors rapidement

prohibitif. Ainsi, si l'on souhaite réaliser un CAN parallèle sur 12 bits, il faudra pas moins de 4095 comparateurs, et on imagine sans peine la taille de la puce et le coût induit. En pratique les convertisseurs parallèle se limitent généralement à une résolution de 8 bits, voire 10 bits. Outre la surface de silicium, la consommation est élevée, et ce d'autant plus que les CAN parallèles de qualité sont réalisés souvent en technologie bipolaire. La réalisation d'un CAN parallèle exige par ailleurs un savoir-faire évident afin d'éviter les erreurs statiques. Une difficulté est également liée au comportement dynamique des comparateurs lorsque le signal d'entrée est situé au voisinage de leur seuil de basculement. Le temps d'établissement d'un comparateur peut alors excéder le temps alloué au décodeur, après application du signal de verrouillage. Ce phénomène, appelé métastabilité, devient de plus en plus fréquent à mesure que la fréquence d'échantillonnage s'élève, et se traduit finalement par des erreurs sporadiques. Enfin, l'impédance d'entrée d'un CAN parallèle est fortement capacitive et de plus non linéaire, puisqu'elle est fonction de l'état des comparateurs. Cela demande un amplificateur de commande large bande supportant cette charge capacitive.

Aujourd'hui, pour obtenir des convertisseurs rapides faisant appel à un nombre réduit d'amplificateurs (par rapport au CAN parallèle pur), des techniques conjointes de recouvrement et d'interpolation (*folding* et *interpolating*) avec prétraitement analogique sont mises en oeuvre.

3.2.1.2 Les convertisseurs pipeline

Dans une architecture pipeline, la conversion est répartie sur au moins deux étages. Notons qu'un CAN parallèle peut tout à fait être pipeliné grâce à la présence de circuits de verrouillage (latches) additionnels. Ceci permet au CAN d'entamer un nouveau cycle de conversion avant que le précédent ne soit achevé. En réalité, le terme pipeline regroupe le plus souvent des convertisseurs dont l'architecture est de type *subranging*, dans laquelle chaque étage prend en charge la conversion d'une partie du signal analogique échantillonné [12]. Dans la version pipelinée à 1 bit par étage, au premier coup d'horloge l'échantillonneur-bloqueur de tête acquiert le signal d'entrée, et celui-ci est comparé à la demi-tension de pleine échelle. Le résultat de la comparaison donne le *MSB* (*Most Significant Bit*). Le processus se répète à l'identique dans l'étage suivant pendant la seconde phase de l'horloge. Les étages impairs sont déclenchés sur la première phase de l'horloge, tandis que les étages pairs le sont sur la seconde. Avec un CAN de résolution N bits, N comparateurs sont suffisants pour réaliser une conversion par cycle, mais avec un temps de latence de N cycles au minimum. Dans une architecture multibit, figure 3.2, un CAN parallèle de faible résolution ($m = 3$ ou 4 bits par exemple) remplace le comparateur. Un bloc générique de cette structure consiste alors en un échantillonneur-bloqueur un CAN et un CNA de même résolution, ainsi qu'un soustracteur. Le premier étage convertit les m bits de poids fort et le ré-

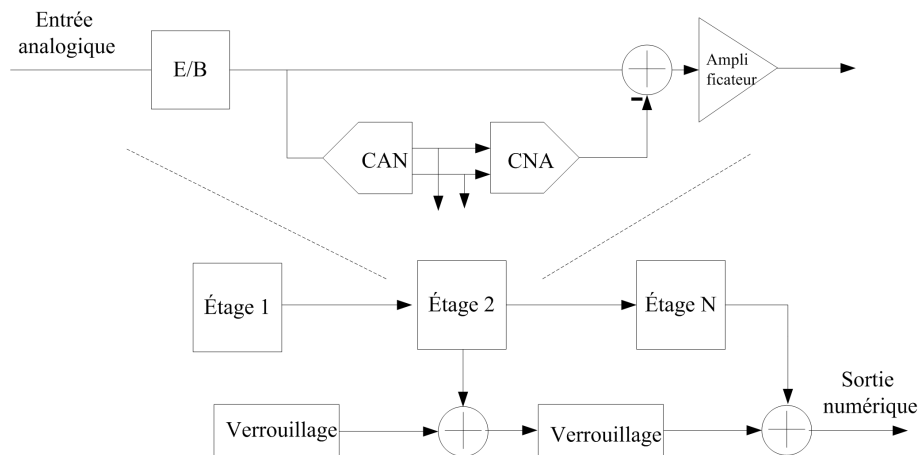


FIG. 3.2 – Convertisseur pipeline.

sultat est envoyé dans le CNA. Ce dernier reconstruit une version grossière du signal analogique, qui est ensuite retranchée à la valeur initialement acquise. Le résidu obtenu est amplifié par 2^m avant d'être délivré à l'étage suivant du pipeline où la procédure se répète. Souvent le CNA est un CNA multiplieur (MDAC) qui fournit le gain entre étage ainsi que la fonction de soustraction. La taille du pipeline dépend de la résolution recherchée. L'investissement matériel réclamé ici est acceptable, puisque chaque CAN parallèle constitutif d'un étage est de résolution modeste. En pratique, quelques bits sont ajoutés afin d'introduire de la redondance et ainsi effectuer la correction des erreurs. Celle-ci est quasi systématique dans les structures pipelinées actuelles, y compris dans la version avec CAN 1 bit par étage, où un comparateur est ajouté pour former ce que l'on appelle communément un étage 1,5 bit.

Le plus souvent réalisés en technologie CMOS, les CAN pipeline présentent un bon compromis entre performance, consommation et coût. Ils fonctionnent à plusieurs dizaines ou centaines de MS/s pour des résolutions de 14 à 10 bits.

3.2.2 Les convertisseurs à haute résolution

3.2.2.1 les convertisseurs Sigma Delta

Les convertisseurs Sigma Delta forment une classe à part de convertisseur analogique numérique appelée convertisseur à suréchantillonnage. En effet, le fait de suréchantillonner l'entrée au-delà de la fréquence de Nyquist, puis de filtrer la sortie numérique en ne conservant que la bande utile du signal permet d'augmenter le rapport signal sur bruit. Cependant, le fait que la fréquence de suréchantillonnage ne peut pas être augmentée indéfiniment constitue la limite de la technique de suréchantillonnage et de filtrage. Ainsi, une autre technique vient alors s'ajouter

au processus de suréchantillonnage : la modulation Sigma Delta [16]. Cette technique consiste à mettre en forme le bruit de quantification afin de réduire son énergie dans la bande utile du signal. Un filtre décimateur vient s'ajouter au modulateur pour filtrer le bruit hors de la bande utile du signal et ramener le fonctionnement du système à la fréquence de Nyquist. Le convertisseur Sigma Delta utilise un quantificateur basse résolution (souvent un seul bit), cadencé à des taux considérablement plus élevés que ceux demandés par le critère de Nyquist. Le quantificateur crée un grand nombre d'échantillons basse résolution qui sont moyennés au cours du temps. En choisissant une quantification sur un bit (deux niveaux logiques), la conception analogique est potentiellement plus simple [17]. Dans le domaine numérique, le filtre décimateur est nécessaire pour reconstruire les données de sortie à la fréquence de Nyquist et retirer le bruit hors bande. La figure 3.3 montre un convertisseur exploitant un bloc modulateur $\Sigma\Delta$ de premier ordre suivi du filtre décimateur. Il comprend un soustracteur, un intégrateur et un

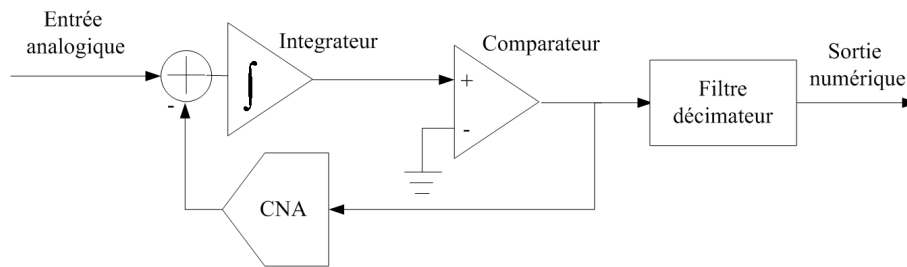


FIG. 3.3 – Convertisseur Sigma Delta.

comparateur. La sortie du comparateur alimente un CNA 1 bit qui ferme la boucle de contre-réaction du modulateur. Ce dernier exerce donc la comparaison du signal d'entrée avec un niveau de référence issu du CNA qui commute entre $\pm V_{ref}$. Le comparateur est cadencé à la fréquence de sur-échantillonnage. Moyennant un gain de boucle suffisant, le modulateur fournit un flux binaire dont la densité de 0 et de 1 est une représentation numérique directe du niveau d'entrée analogique. Les erreurs de quantification au sein du modulateur limitent la plage dynamique. Pour un CAN $\Sigma\Delta$, l'augmentation de la résolution est obtenue par l'obtention d'un équilibre entre le taux de sur-échantillonnage, l'ordre du modulateur, et la résolution du quantificateur [16] [18].

Plus l'ordre du modulateur est élevé, meilleur sera le SNR . Le problème majeur lors de la conception de tels convertisseurs est que dès qu'un ordre élevé (supérieur à deux) est visé, il se pose des problèmes de stabilité [16]. En effet le quantificateur est un élément non linéaire dont la modélisation n'est plus assez précise. En pratique, au détriment du SNR , plus l'ordre augmente plus la dynamique du signal analogique d'entrée doit diminuer afin de conserver la stabilité.

3.2.2.2 Les convertisseurs par approximations successives

Les convertisseurs par approximations successives réalisent la conversion par comparaison en plusieurs étapes. A chaque étape, le résultat de cette comparaison est stocké dans un registre, jusqu'à l'obtention du résultat final. Le principe consiste à comparer la tension du signal d'entrée avec une tension issue d'un registre après conversion numérique analogique (CNA). Le schéma bloc de la figure 3.4 montre le fonctionnement d'un tel convertisseur. Au départ le *MSB* du

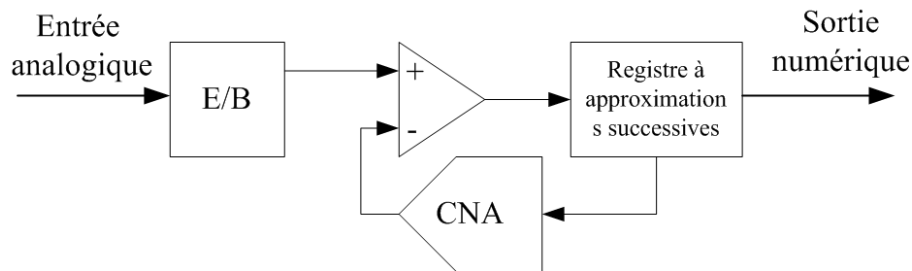


FIG. 3.4 – Convertisseur par approximations successives.

registre est mis à 1 et les autres bits sont à 0. Ainsi, la tension en sortie du CNA est égale à la moitié de la pleine échelle. Si la tension d'entrée est inférieure à cette valeur, le *MSB* est mis à 0. A son tour, le bit suivant sera mis à 1 et une nouvelle comparaison avec le signal d'entrée sera faite, jusqu'au dernier bit du registre. Une résolution de N bits implique une durée de conversion minimale de N cycles d'horloge. Son principal inconvénient est donc le temps de conversion, qui limite ses performances à quelques MS/s environ. En dépit de sa lenteur, ce convertisseur présente l'avantage d'être simple à réaliser et d'avoir une surface très réduite. De plus, il permet d'atteindre des résolutions assez considérables avec de bonnes performances en termes de *NLD* et *NLI*.

3.2.3 De l'architecture d'un CAN vers les TIADC

L'étude succincte des différentes architectures de conversion analogique numérique montre que la précision et la vitesse sont deux critères difficiles à combiner. Le tableau 3.1 montre que, quelque soit l'architecture du CAN, un des deux paramètres cités est pénalisé. De plus, la consommation de puissance est un critère critique pour plusieurs applications, particulièrement lorsqu'il s'agit d'augmenter la fréquence d'échantillonnage maximale. C'est le cas des convertisseurs parallèles où le réseau de résistance et le nombre de comparateurs font que la consommation de puissance augmente considérablement.

Concrètement, les couples vitesse/résolution proposés conviennent à la plupart des applications en rapport avec l'instrumentation, la vidéo, l'imagerie médicale et les télécommunications :

Architecture	Vitesse maximale	Résolution	Consommation
Parallèle	Des centaines de MS/s	8 à 10	↗↗
Pipeline numérique	Des dizaines de MS/s	12 à 14	↗
Sigma Delta	5 MS/s	16 à 20	↘↘
SAR	Une dizaines de MS/s	14 à 16	↘

TAB. 3.1 – Performance d'un CAN.

échantillonnage en FI, boucles de prédistorsion numérique pour la linéarisation des PA des stations de base 3G, etc. Cependant, dans certains domaines, le CAN est encore le maillon faible de la chaîne de traitement du signal. C'est souvent le cas dans les récepteurs RF fonctionnant en mode large bande ou multibande ou dans la radio logicielle, où il y a un intérêt à numériser le signal en bande RF au plus près de l'antenne. Le problème est que la dynamique du signal est importante (surtout dans le cas du multistandard) et par conséquent un nombre de bits important est exigé pour numériser le signal. C'est la raison pour laquelle la numérisation du signal se fait toujours en FI.

Pour augmenter la fréquence d'échantillonnage, l'entrelacement temporel des CAN est une méthode conceptuellement simple pour multiplier la cadence d'échantillonnage. Cependant, ce concept à été peu industrialisé à cause des problèmes de disparité entre les différents convertisseurs entrelacés dans le temps [19].

3.3 Principe des CAN entrelacés dans le temps

Pour concevoir un CAN de résolution 7 bits à 4 MS/s de fréquence d'échantillonnage, Black et Hodges en 1980, démontrent qu'un entrelacement de deux convertisseurs à une fréquence d'échantillonnage de 2 MS/s est une solution possible qui occupe moins de surface qu'un seul convertisseur parallèle ayant les mêmes caractéristiques [19]. Le gain en surface de silicium n'est pas le seul bénéfice de ce nouveau concept. L'entrelacement dans le temps offre une méthode simple pour multiplier le taux d'échantillonnage. Théoriquement, et pour une technologie donnée, l'entrelacement dans le temps de plusieurs convertisseurs ne présente pas de limite et peut donc atteindre des taux d'échantillonnages infinis. Mais, réellement, la puissance dissipée, la taille et la bande passante limitent le nombre de convertisseurs à utiliser et par la suite le taux d'échantillonnage résultant.

Ce concept a donc été introduit pour des systèmes nécessitant un taux d'échantillonnage élevé. Par conséquent, il est surtout utilisé dans l'industrie du test et de la mesure, particulièrement pour les oscilloscopes numériques : par exemple l'oscilloscope Hewlett-Packard HP54720 utilise

4 CAN cadencés chacun à 500 MS/s pour atteindre un taux maximum de 2 GS/s. Avec le même type de convertisseurs, l'entrelacement de 16 CAN permet d'obtenir un taux de 8 GS/s et ce principe a été adopté pour l'oscilloscope HP54722, [20]. Dans la famille d'oscilloscopes d'Infenium, les concepteurs d'Agilent Labs ont introduit un convertisseur de résolution 8 bits à 20 GS/s, [21]. Atmel a mis sur le marché un convertisseur dual AT84AD001 de résolution 8 bits qui fonctionne 2 GS/s pour les applications RADAR, [22].

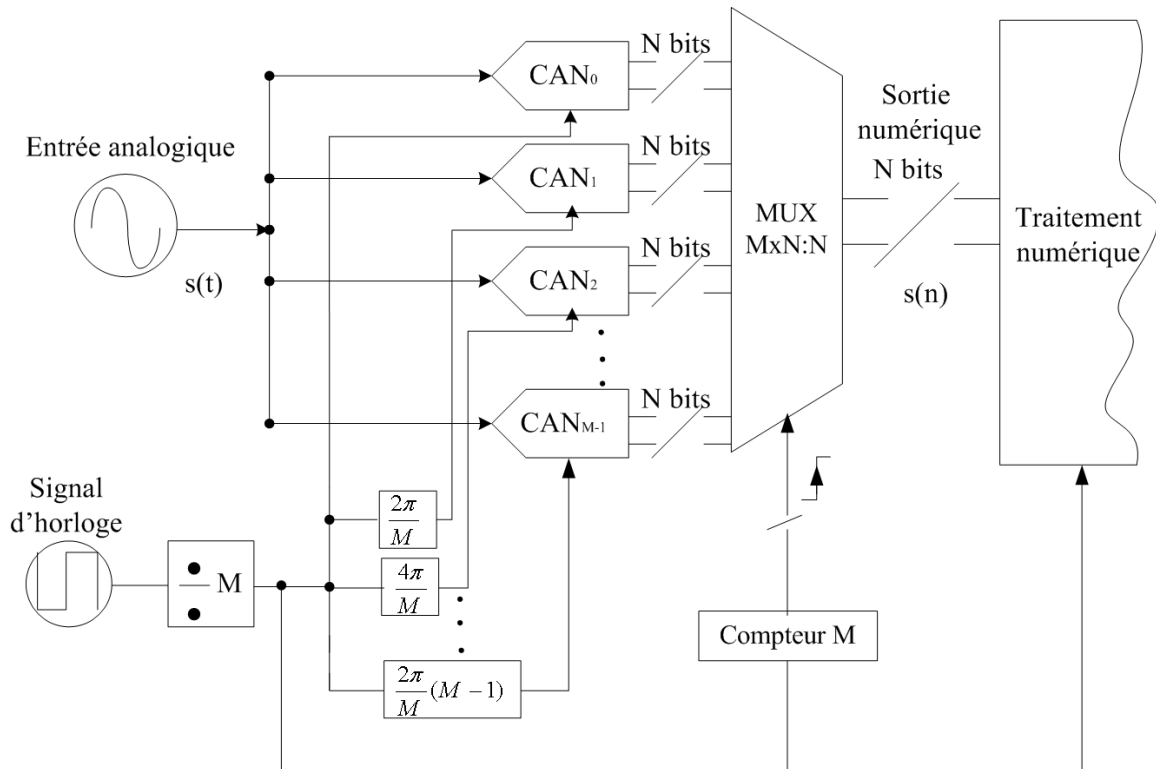


FIG. 3.5 – Principe de la conversion par entrelacement.

La figure 3.5 présente une architecture complète d'un TIADC contenant M convertisseurs. Les blocs essentiels d'une telle structure sont : M CAN en parallèles, un démultiplexeur analogique en entrée et un multiplexeur numérique à la sortie. Chaque convertisseur échantillonne le même signal analogique à une fréquence équivalente au $\frac{1}{M}$ de la fréquence d'échantillonnage de toute la structure. Durant le processus de conversion, le démultiplexeur, jouant le rôle de la circuiterie de redistribution de l'horloge, fournissant à chaque convertisseur des signaux d'horloge décalés d'une période divisée par M . Le même signal d'entrée attaque les différents convertisseurs. Après l'étape de conversion, le multiplexeur, piloté par un compteur de front montants des différentes horloges, regroupe les données des convertisseurs d'une façon cyclique. Le taux d'échantillonnage du système est ainsi M fois plus important que celui de chaque convertisseur.

En comparant les convertisseurs entrelacés actuels au convertisseur de résolution 7 bits à une fréquence d'échantillonnage de 4 MS/s proposé par Black en 1980, nous remarquons qu'il y a une énorme évolution au niveau de la fréquence d'échantillonnage. En effet, 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 8 bits.

L'augmentation du taux d'échantillonnage du système se fait donc au dépend de la précision. En effet, les erreurs statiques ainsi que les erreurs d'ouverture citées dans le chapitre 2 sont des erreurs constantes pour un convertisseur donné mais aléatoires d'un composant à un autre. Elles sont engendrées par les défaillances du processus de fabrication, le vieillissement du composant ou les conditions d'utilisation antérieures.

Pour deux CAN, 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. Dans la suite, l'origine et les aspects des disparités des erreurs sont explicitement introduits. Les formules mathématiques induites montrent qu'il s'agit parfois d'erreurs non déterministes. L'influence de ces erreurs sur les paramètres dynamiques d'un CAN est aussi présentée.

3.4 Analyse des effets de disparité des erreurs

Dans les sections suivantes, les différentes erreurs d'un TIADC comprenant M convertisseurs sont étudiées séparément. Ces erreurs sont aléatoires d'un composant à un autre et sont supposées constantes au cours du cycle de conversion. Une analyse d'un cas particulier d'un TIADC formé par deux CAN est présentée afin d'évaluer la dégradation du SNR à la sortie de la structure entrelacée.

Le signal d'entrée à convertir est décrit par l'équation suivante :

$$s(t) = A \sin(2\pi f_0 t + \varphi) \quad (3.1)$$

où A est l'amplitude du signal à l'entrée du convertisseur, f_0 la fréquence du signal d'entrée et φ la phase du signal d'entrée. Soit l , $l \in [1, M]$, l'ordre du CAN utilisé dans le TIADC. Les instants d'échantillonnage de chaque CAN sont exprimés par la relation suivante :

$$T_l(k) = kMT_s + lT_s \quad (3.2)$$

k est un entier positif qui représente l'ordre des échantillons, T_s est la période d'échantillonnage de chaque convertisseur. Ainsi, en supposant négligeable le bruit de quantification, la sortie d'un convertisseur idéal de la structure entrelacée s'exprime par :

$$y_l(t) = A \sin(2\pi f_0 t + \varphi) \times \overbrace{\sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s)}^{p(t)} \quad (3.3)$$

3.4.1 Influence des erreurs statiques

3.4.1.1 Influence de l'erreur d'offset

L'erreur d'offset est une constante additive au signal d'entrée. Elle est notée O_l pour le convertisseur l et est modélisée mathématiquement conformément à la figure 3.6. En présence d'erreur

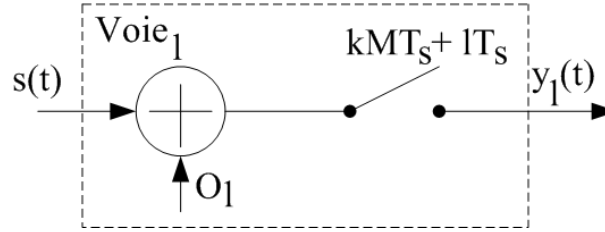


FIG. 3.6 – Modèle mathématique de l'erreur d'offset.

d'offset, la sortie sortie $y_l(t)$ de la figure 3.6 est exprimée par la relation suivante :

$$y_l(t) = (A \sin(2\pi f_0 t + \varphi) + O_l) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.4)$$

L'erreur due à l'offset à la sortie de chaque CAN, $eof f_l(t)$, correspond à la différence entre les équations 3.4 et 3.3 :

$$eof f_l(t) = O_l \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.5)$$

L'analyse temporelle de l'erreur d'offset pour M convertisseurs est complexe. Pour cette raison, nous utilisons la Transformée de Fourier (TF) pour avoir une autre vue du signal d'erreur. Selon l'équation 3.5, $eof f_l(t)$ est le produit de la constante d'offset, O_l par la fonction peigne de Dirac espacé $p(t)$. La Transformée de Fourier de $eof f_l(t)$ correspond au produit de convolution de la Transformée de Fourier de O_l et du peigne de Dirac. La Transformée de Fourier la constante O_l est :

$$TF(O_l) = O_l(f) = \int_{-\infty}^{+\infty} O_l \exp^{-2j\pi ft} dt = O_l \delta(f) \quad (3.6)$$

Celle de la fonction peigne de Dirac est donnée par [23] [24] :

$$\begin{aligned} TF(p(t)) = P(f) &= \int_{-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \exp^{-2j\pi ft} dt \\ &= \frac{1}{MT_s} \sum_{k=-\infty}^{+\infty} \delta\left(f - \frac{kf_s}{M}\right) \exp^{-jkl\frac{2\pi}{M}} \end{aligned} \quad (3.7)$$

La Transformée de Fourier de $eof f_l(t)$, produit de convolution des fonctions $P(f)$ et $O_l(f)$, s'écrit sous la forme :

$$TF(eof f_l(t)) = TF(O_l \times p(t)) = \frac{1}{2\pi} (O_l(f) * P(f)) \quad (3.8)$$

ou encore :

$$TF(eof f_l(t)) = EOF F_l(f) = \sum_{k=-\infty}^{+\infty} \frac{O_l}{2\pi MT_s} \delta\left(f - \frac{kf_s}{M}\right) \exp^{-jkl\frac{2\pi}{M}} \quad (3.9)$$

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

$$EOFF(f) = \sum_{l=1}^M \sum_{k=-\infty}^{+\infty} \frac{O_l}{2\pi MT_s} \delta\left(f - \frac{kf_s}{M}\right) \exp^{-jkl\frac{2\pi}{M}} \quad (3.10)$$

L'équation 3.10 montre l'apparition de plusieurs raies non-harmoniques : la composante DC et des composantes localisées à $\frac{kf_s}{M}$. Pour un TIADC comprenant deux CAN, l'équation 3.10 s'écrit sous la forme suivante :

$$EOFF(f) = \frac{1}{4\pi T_s} \sum_{k=-\infty}^{+\infty} \delta\left(f - \frac{kf_s}{2}\right) [O_2 + (-1)^k O_1] \quad (3.11)$$

Le spectre de la sortie du TIADC composé par deux CAN présenté dans la figure 3.7 confirme les formules mathématiques. Deux raies non-harmoniques s'ajoutent au spectre : une raie d'offset et une raie à la fréquence $\frac{f_s}{2}$. La composante DC et celle à la fréquence de Nyquist dégradent les performances spectrales du TIADC. En effet, si les deux convertisseurs étaient parfaits, le SNR résultant, ne comprenant que la contribution du bruit de quantification, devrait être conforme à la relation de l'équation (2.22) puisque les composantes continues n'interviennent pas dans l'analyse spectrale du signal.

Pour calculer la contribution de la disparité d'offset sur les paramètres spectraux, tout en restant sur deux dimensions, nous supposons que l'un des deux offsets est nul. L'offset est normalisé par rapport à l'amplitude du signal d'entrée. Le rapport signal sur bruit s'écrit donc sous la forme

$$SNR = 20 \log_{10} \left(\frac{V_{eff}}{B_q} \right) \quad (3.12)$$

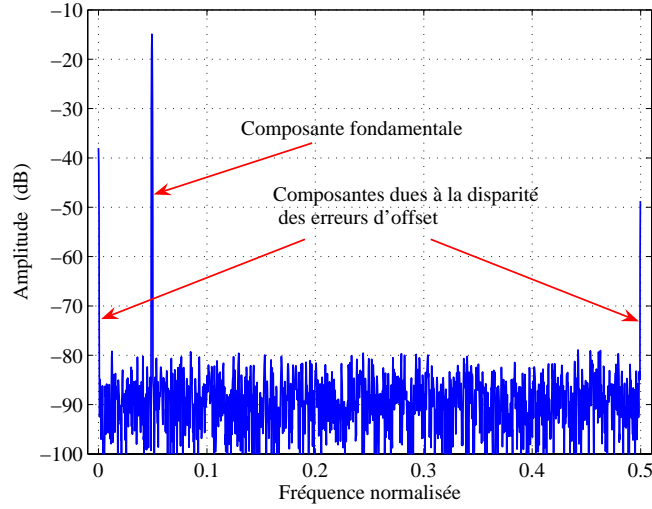


FIG. 3.7 – Effet du disparité d'offset sur le spectre d'un TIADC.

où V_{eff} est la valeur efficace du signal de sortie du TIADC et B_q représente la somme du bruit de quantification, des composantes harmoniques et des composantes non harmoniques. Le SNR résultant, dépend donc de la résolution des convertisseurs et aussi de l'offset du premier convertisseur (l'offset du deuxième est supposé nul). On montre qu'il s'exprime de façon analytique par la relation suivante :

$$SNR = 6.02N + 1,76 \overbrace{-10 \log_{10}(3 \times 2^{2N-1} (O_2 - O_1)^2 + 1)}^{perte} \quad (3.13)$$

Pour $O_1 = 0$ l'équation (3.13) tend vers l'équation (2.22). De plus, pour valider cette étude, nous avons comparé cette formule avec des résultats de simulation d'un convertisseur entrelacé sur lesquelles nous avons appliqué une analyse spectrale permettant l'estimation du SNR . Les résultats sont présentés sur la figure 3.8.

D'après cette figure, pour un offset égale à zéro, la perte est nulle. Cette perte en SNR croit exponentiellement avec l'augmentation de la valeur absolue de l'offset normalisé par rapport à l'amplitude du signal d'entrée. La variation du SNR résultant ne dépend ni de la fréquence du signal d'entrée ni de son amplitude. Ceci revient au fait que l'erreur d'offset est un bruit constant (pour un seul convertisseur), additif et non corrélé avec le signal d'entrée.

3.4.1.2 Influence de l'erreur de gain

L'erreur de disparité des gains dans le TIADC est traitée par la même façon que celle de l'offset. Par principe, l'erreur de gain est une variable aléatoire, constante tout au long du cycle de

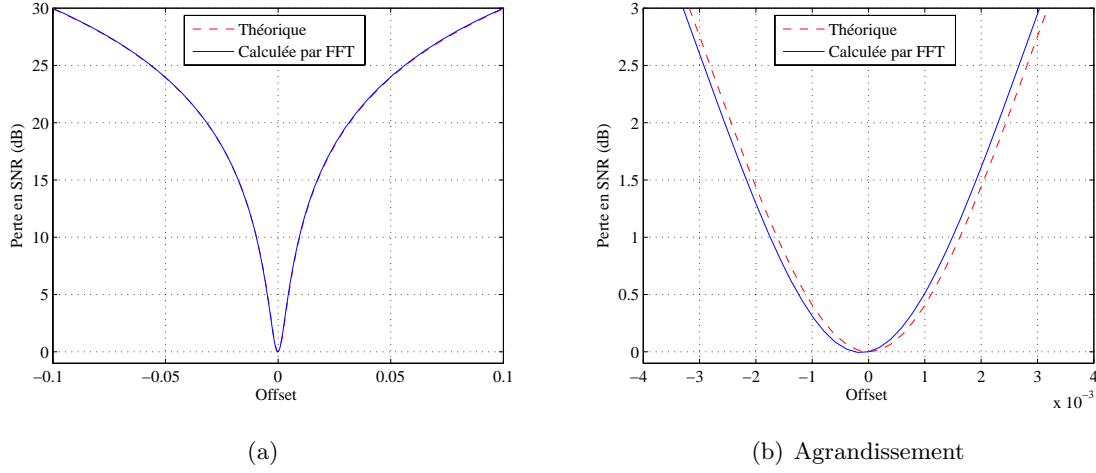


FIG. 3.8 – Effet du disparité d'offset sur le SNR d'un TIADC.

conversion, qui est multipliée à l'amplitude A du signal d'entrée. Elle est modélisée mathématiquement conformément à la figure 3.9. L'erreur de gain est proche de 1 et elle modélisée par une

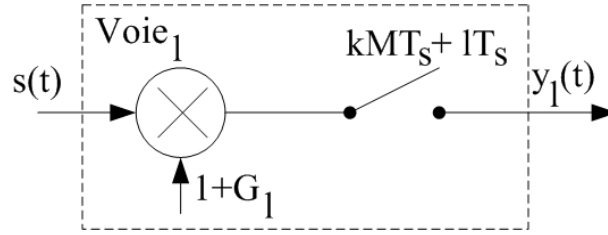


FIG. 3.9 – Modèle mathématique de l'erreur de gain.

constante de la forme $(1 + G_l)$ pour le convertisseur l . En présence d'erreur de gain, la sortie $y_l(t)$ de la figure 3.9 est exprimée par la relation suivante :

$$y_l(t) = (A(1 + G_l) \sin(2\pi f_0 t + \varphi)) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.14)$$

L'erreur due au gain à la sortie de chaque CAN, $egain_l(t)$, correspond à la différence entre les équations 3.14 et 3.3 :

$$egain_l(t) = \overbrace{AG_l \times \sin(2\pi f_0 t + \varphi)}^{gain_l(t)} \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.15)$$

De la même manière que pour l'étude de la disparité d'offset, nous utilisons la Transformée de Fourier pour mettre l'accent sur l'influence des disparités de gain dans un TIADC comprenant

M CAN. L'équation 3.15 se présente comme le produit de la fonction peigne de Dirac espacé, dont la Transformée de Fourier est donnée dans l'équation 3.7, avec un signal sinusoïdal. La Transformée de Fourier du signal sinusoïdal $gain_l(t)$ de l'équation 3.15 est de la forme :

$$TF(gain_l(t)) = GAIN_l(f) = \frac{jAG_l}{2} (\delta(f - f_0) - \delta(f + f_0)) \quad (3.16)$$

La Transformée de Fourier de $egain_l(t)$, produit de convolution des fonctions $P(f)$ et $GAIN_l(f)$, s'écrit sous la forme :

$$EGAIN_l(f) = \sum_{k=-\infty}^{+\infty} \frac{jAG_l}{4\pi MT_s} \left[\delta\left(f - \frac{kf_s}{M} + f_0\right) - \delta\left(f - \frac{kf_s}{M} - f_0\right) \right] \exp^{-jkl\frac{2\pi}{M}} \quad (3.17)$$

Finalement la Transformée de Fourier de l'erreur engendrée par la disparité des erreurs de gain dans un TIADC composé de M CAN est exprimée par :

$$EGAIN(f) = \sum_{l=1}^M \sum_{k=-\infty}^{+\infty} \frac{jAG_l}{4\pi MT_s} \left[\delta\left(f - \frac{kf_s}{M} + f_0\right) - \delta\left(f - \frac{kf_s}{M} - f_0\right) \right] \exp^{-jkl\frac{2\pi}{M}} \quad (3.18)$$

L'équation 3.18 montre l'existence de raies non-harmoniques localisées aux fréquences $\frac{kf_s}{M} - f_0$ et de son repliement spectral aux fréquences $\frac{kf_s}{M} + f_0$. Pour le cas de deux CAN, l'équation 3.18 s'écrit sous la forme :

$$EGAIN(f) = \sum_{k=-\infty}^{+\infty} \frac{jA}{8\pi T_s} \left[\delta\left(f - \frac{kf_s}{2} + f_0\right) - \delta\left(f - \frac{kf_s}{2} - f_0\right) \right] (G_2 + (-1)^k G_1) \quad (3.19)$$

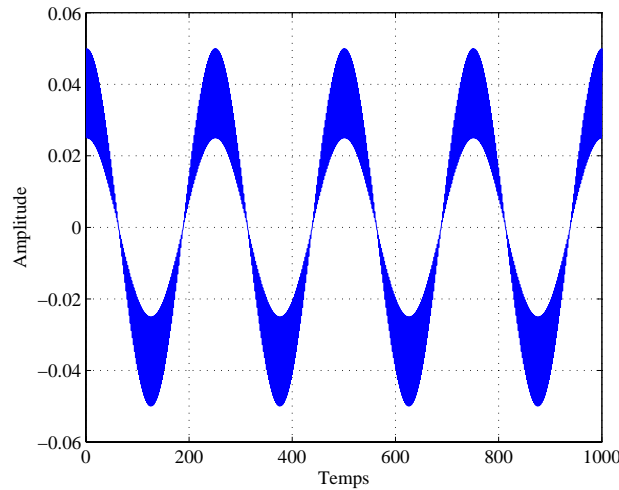


FIG. 3.10 – Erreur temporelle de la disparité de gain

Contrairement au cas de disparité d'offset, l'amplitude de l'erreur de gain est modulée par la fréquence du signal d'entrée comme dans le cas d'une modulation AM . La figure 3.10 représente

l'erreur temporelle due à la disparité de gain. L'analyse spectrale du signal de sortie du TIADC avec deux erreurs de gain différentes est montrée dans la figure 3.11. L'amplitude de la raie

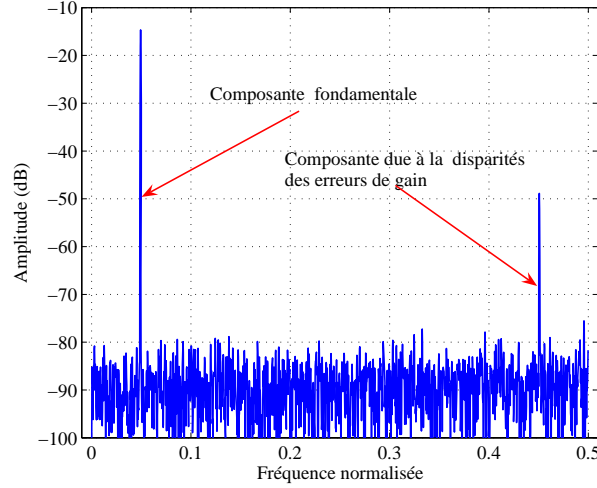


FIG. 3.11 – Effet de la disparité de gain sur le spectre d'un TIADC.

spectrale située à la fréquence $\frac{f_s}{2} - f_0$ dépend des valeurs de gain des deux convertisseurs. Pour les mêmes valeurs de gain, l'amplitude de cette raie dépend aussi de l'amplitude A du signal d'entrée. La perte en SNR est exprimée dans l'équation suivante :

$$SNR = 6.02N + 1,73 - \overbrace{10 \log_{10} \left(3 \times 2^{2N-1} \left(\frac{G_2 - G_1}{2} \right)^2 + 1 \right)}^{perte} + 20 \log_{10} \frac{G_2 + G_1}{2} \quad (3.20)$$

De la même façon que dans le cas de l'offset, nous avons comparé cette formule avec des résultats de simulation. Cette comparaison est donnée à travers la figure 3.12.

Contrairement à la disparité d'offset où la composante non harmonique due au non appariement ne dépend que la fréquence d'échantillonnage, la disparité des erreurs de gain dépend du signal d'entrée. La localisation de cette composante varie en fonction de la fréquence d'entrée. De plus, la composante due à la disparité d'offset reste figée, par contre celle de la disparité de gain varie en fonction du canal sélectionné.

3.4.1.3 Erreurs de non-linéarité

Pour un convertisseur analogique numérique, la non-linéarité différentielle NLD peut s'écrire sous la forme :

$$NLD(k) \approx \frac{N_k}{N_t \times P(k)} - 1 \quad (3.21)$$

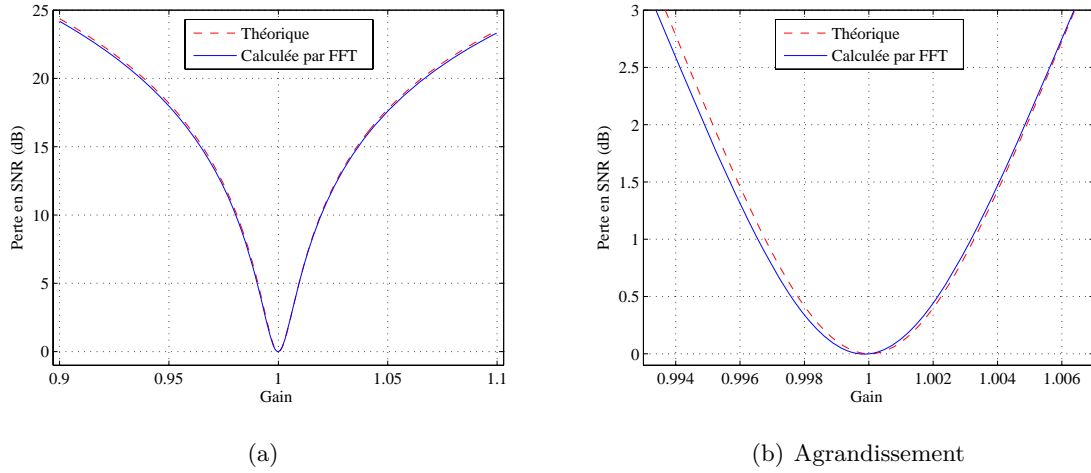


FIG. 3.12 – Effet de la disparité de gain sur le SNR d'un TIADC.

où N_k est le nombre d'apparition du code k , N_t est le nombre total d'échantillons et $P(k)$ est la probabilité théorique du code k [25]. Pour un TIADC composé de deux CAN, chacun d'entre eux convertit $\frac{N_t}{2}$ échantillons. Si la fréquence du signal d'entrée et la fréquence d'échantillonnage sont choisies de manière à éviter des séquences répétitives du même échantillon [26], les deux convertisseurs auront la même probabilité $P(k)$ pour un code k . Mais, puisque les deux CAN ont une caractéristique de transfert différente, le nombre d'occurrences du code k est différent. Soit N_{k1} le nombre d'occurrence pour le premier convertisseur et $N_{k2} = N_k - N_{k1}$ celui du deuxième convertisseur. La NLD du TIADC peut se mettre sous la forme :

$$NLD(k) = \frac{N_k}{N_t \times P(k)} - 1 = \frac{N_{k1} + N_{k2}}{\frac{N_t}{2}P(k) + \frac{N_t}{2}P(k)} - 1 \quad (3.22)$$

Ceci est équivalent à

$$NLD(k) = \frac{1}{2} \left\{ \frac{N_{k1}}{\frac{N_t}{2}P(k)} - 1 + \frac{N_{k2}}{\frac{N_t}{2}P(k)} - 1 \right\} \quad (3.23)$$

Soit $NLD_1(k)$ et $NLD_2(k)$ les non-linéarités différentielles de chaque convertisseur, l'équation (3.23) devient :

$$NLD(k) = \frac{NLD_1(k) + NLD_2(k)}{2} \quad (3.24)$$

Ce résultat s'applique aussi pour la non-linéarité intégrale NLI :

$$NLI(k) = \sum_{j=1}^k NLD(j) = \frac{1}{2} (NLI_1(k) + NLI_2(k)) \quad (3.25)$$

Ainsi, la NLD et la NLI du TIADC s'exprime en fonction de la NLD et de la NLI de chaque convertisseur. Les erreurs de la NLD et de la NLI d'un convertisseur entrelacé peuvent être inférieures à celles des deux convertisseurs et sont, dans pire des cas, inférieures à celles du convertisseur présentant les erreurs de non-linéarités les plus importantes. Ce constat a été validé par mesures dans [27].

Pour vérifier cette étude, la non-linéarité différentielle d'une structure entrelacée de deux convertisseurs est calculée par la méthode des histogrammes. La figure 3.13 montre les erreurs de la NLD de chaque convertisseurs ainsi que celle du TIADC. Conformément à ce que a été montré

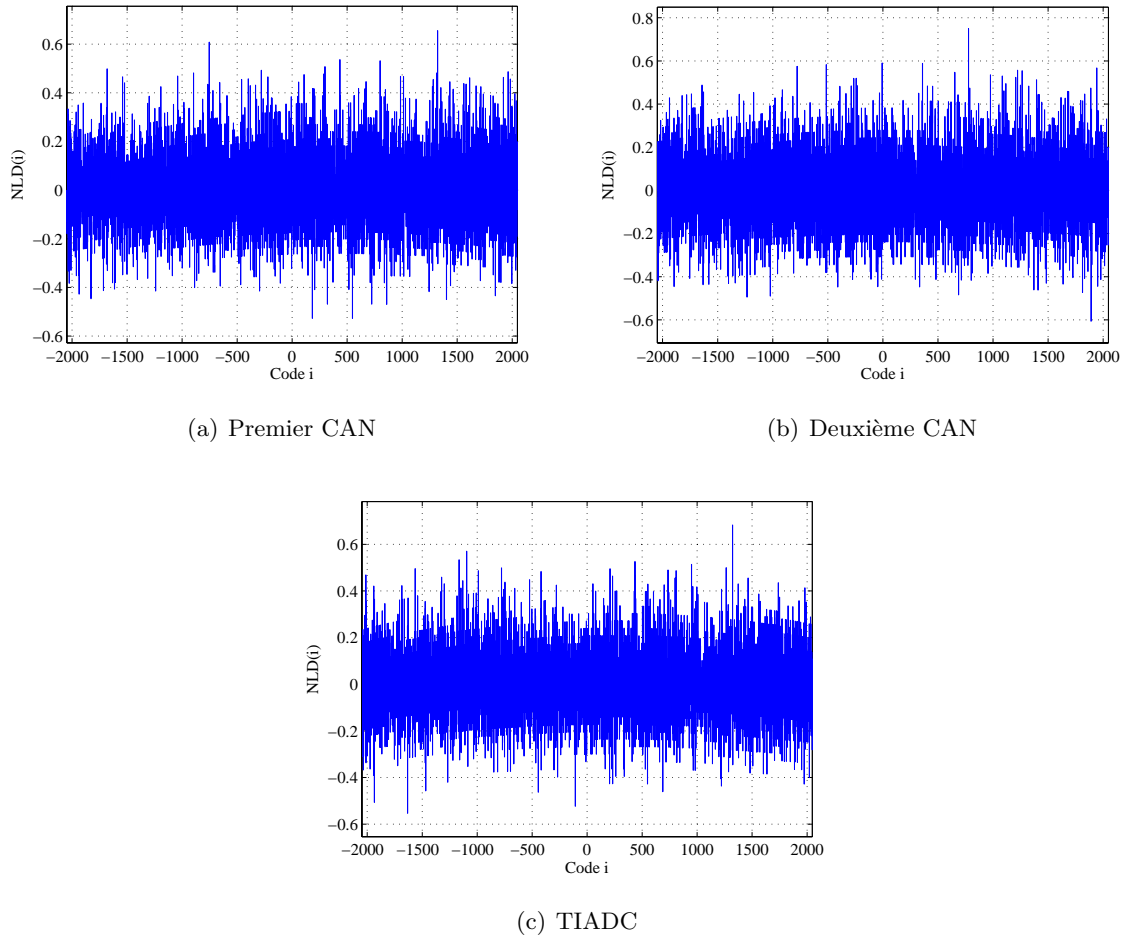


FIG. 3.13 – Erreur de la NLD pour chaque CAN et pour le TIADC.

dans [26] et dans [27], la NLD d'un TIADC est inférieure à celle du convertisseur présentant les

erreurs de non-linéarité les plus importantes. Le tableaux 3.2 montre les différents paramètres extraits de l'analyse de la *NLD* de chaque convertisseur et du convertisseur entrelacé.

	Moyenne	Écart type	Peak-min	Peak-max	Peak-to-peak
CAN 1	$1.0279.10^{-6}$	0.1790	-0.6068	0.7495	1.3563
CAN 2	$1.0999.10^{-5}$	0.1642	-0.5275	0.6561	1.1836
TIADC	$-3,1177.10^{-5}$	0.1625	-0.5534	0.6829	1.2363

TAB. 3.2 – Paramètres de non-linéarité différentielle de chaque CAN et du TIADC.

Les paramètres **Peak-min**, **Peak-max** et **Peak-to-peak** montrent respectivement l'amplitude minimale, maximale et crête à crête de la *NLD*. Tous ces paramètres, en plus de l'écart type, sont plus importants dans le premier CAN que dans dans le TIADC.

Une deuxième étude sur les erreurs de non-linéarités des TIADC est détaillée dans [28]. Il a été démontré que les résultats annoncés dans [26] et dans [27] ne sont que partiellement vrais. En effet le comportement des erreurs de non-linéarités est dynamique et par la suite l'application des sommes dans les équations de 3.22 à 3.25 doit se faire avec plus de précaution. Il faut donc tenir compte des raies de disparité des erreurs de non linéarités. Des simulations ont été réalisées pour évaluer la contribution spectrale des erreurs de non-linéarité. La figure 3.14 montre les spectres de sortie de chaque convertisseur présentant des erreurs de non-linéarité différentes ainsi que le spectre du convertisseur entrelacé. Les erreurs de non-linéarité sont modélisées par un polynôme d'ordre 5 multiplicatif au signal d'entrée. Les spectres de la figure 3.14 montrent l'apparition de raies harmoniques de deuxième et troisième ordre pour chaque convertisseur. Dans le cas du TIADC, le spectre de la figure 3.14.(c) montre l'apparition des raies non-harmoniques supplémentaires d'ordres supérieures.

3.4.2 Influences des erreurs dynamiques

3.4.2.1 Influence de l'erreur de jitter

Dans le cas d'un TIADC, chaque convertisseur peut avoir deux erreurs de jitter différentes. Les deux erreurs ont deux distributions gaussiennes avec une moyenne nulle et un écart type différent. L'effet aléatoire du jitter fait qu' à la sortie du TIADC, les délais obtenues en sortie de chaque convertisseur sont par conséquent aléatoires. Donc, contrairement aux cas de l'offset et du gain, où l'erreur est déterministe, et les répercutions sur le spectre apparaissent sous forme de composantes non harmoniques, l'effet des erreurs de jitter sur le spectre de sortie du TIADC se traduit par un bruit supplémentaire qui vient s'ajouter au bruit de quantification [29], [30].

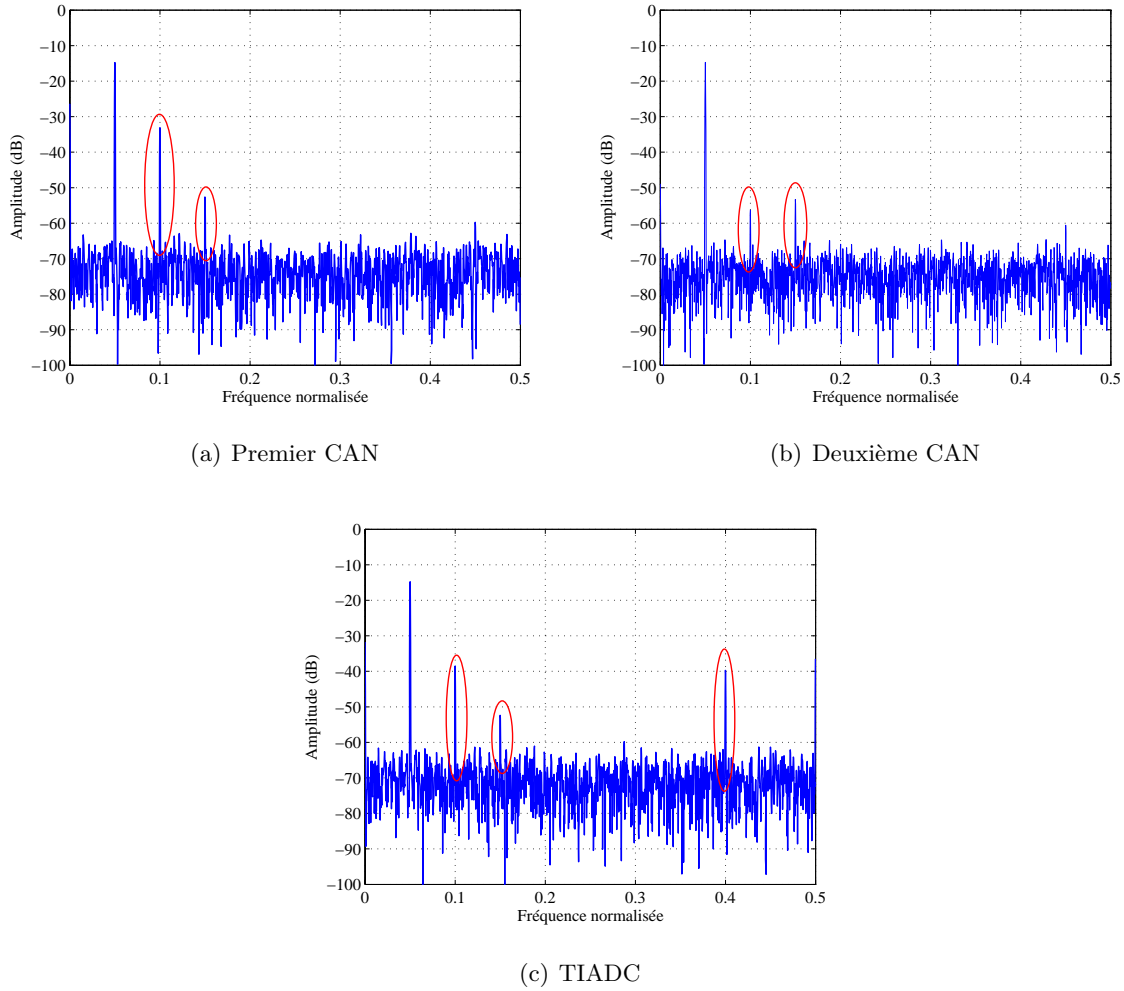


FIG. 3.14 – Effet des erreurs de non-linéarité pour chaque CAN et pour le TIADC.

3.4.2.2 Influence de l'erreur de retard d'ouverture

Contrairement à l'erreur de jitter, le retard d'ouverture est due aux défauts de conception, [31]. Les causes sont diverses, mais nous pouvons citer la disparité sur le circuit, un layout asymétrique du générateur d'horloge et les perturbations sur les tensions d'alimentation. Pendant le même cycle de conversion, le retard d'ouverture est un délai constant.

Supposant que les CAN possèdent chacun une erreur de retard d'ouverture différente δt_l . Ces erreurs modélisées mathématiquement conformément à la figure 3.15. La sortie du modèle de la figure 3.15 s'écrit donc sous la forme :

$$y_l(t) = (A \sin(2\pi f_0(t + \delta t_l) + \varphi)) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.26)$$

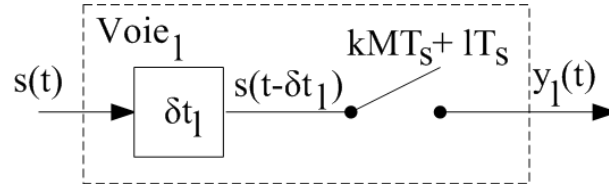


FIG. 3.15 – Modèle mathématique de l'erreur de retard d'ouverture.

Le développement de la fonction $\sin(a + b)$ de l'équation 3.26 donne :

$$y_l(t) = (A \sin(2\pi f_0 t + \varphi) \cos(2\pi f_0 \delta t_l)) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \\ + (A \cos(2\pi f_0 t + \varphi) \sin(2\pi f_0 \delta t_l)) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.27)$$

Étant donné que la valeur de l'erreur du retard d'ouverture est très faible par rapport à la période du signal d'entrée, le développement limité des fonctions *sinus* et *cosinus* au voisinage de δt_l simplifie l'équation (3.27) :

$$y_l(t) \cong A \sin(2\pi f_0 t + \varphi) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \\ + A2\pi f_0 \delta t_l \cos(2\pi f_0 t + \varphi) \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.28)$$

L'erreur due au retard d'ouverture à la sortie de chaque CAN, $eskew_l(t)$, est la différence entre les équations 3.28 et 3.3 :

$$eskew_l(t) = + \overbrace{A2\pi f_0 \delta t_l \cos(2\pi f_0 t + \varphi)}^{skew_l(t)} \times \sum_{k=-\infty}^{+\infty} \delta(t - kMT_s - lT_s) \quad (3.29)$$

De la même manière que pour l'étude de la disparité d'offset et de gain, nous utilisons la Transformée de Fourier pour mettre l'accent sur l'influence des disparités du retard d'ouverture dans un TIADC comprenant M CAN. La Transformée de Fourier de la composante cosinusoidale de l'équation (3.29) est de la forme :

$$TF(skew_l(t)) = SKEW_l(f) = \frac{A2\pi f_0 \delta t_l}{2} (\delta(f + f_0) + \delta(f - f_0)) \quad (3.30)$$

La Transformée de Fourier de $eskew_l(t)$ est le produit de convolution des fonctions $P(f)$, détaillée dans l'équation 3.7, et $SKEW_l(f)$. Elle s'écrit sous la forme :

$$ESKEW_l(f) = \sum_{k=-\infty}^{+\infty} \frac{A\pi f_0 \delta t_l}{2\pi MT_s} \left[\delta\left(f - \frac{kf_s}{M} - f_0\right) + \delta\left(f - \frac{kf_s}{M} + f_0\right) \right] \exp^{-jkl\frac{2\pi}{M}} \quad (3.31)$$

Finalement, la Transformée de Fourier de l'erreur engendrée par le retard d'ouverture dans un TIADC composé de M CAN s'écrit sous la forme :

$$ESKEW(f) = \sum_{l=1}^M \sum_{k=-\infty}^{+\infty} \frac{Af_0 \delta t_l}{2MT_s} \left[\delta \left(f - \frac{kf_s}{M} - f_0 \right) + \delta \left(f - \frac{kf_s}{M} + f_0 \right) \right] \exp^{-jkl \frac{2\pi}{M}} \quad (3.32)$$

L'équation 3.32 montre l'existence de raies non-harmoniques localisées aux fréquences $\frac{kf_s}{M} - f_0$ et de leur repliement spectrale aux fréquences $\frac{kf_s}{M} + f_0$. Pour le cas de deux CAN, l'équation 3.32 s'écrit sous la forme :

$$ESKEW(f) = \frac{Af_0}{4T_s} \sum_{k=-\infty}^{+\infty} \left[\delta \left(f - \frac{kf_s}{2} - f_0 \right) + \delta \left(f - \frac{kf_s}{2} + f_0 \right) \right] (\delta t_2 + (-1)^k \delta t_1) \quad (3.33)$$

D'un point de vue temporel (figure 3.16) la période de cette erreur dépend de la fréquence d'entrée. L'amplitude est fonction des retards δt_1 et δt_2 , de la fréquence d'entrée et de l'amplitude du signal d'entrée. Enfin, au niveau de la phase, cette erreur est retardée de $\frac{\pi}{2}$ par rapport à celle de la disparité de gain. Il s'agit donc de deux erreurs orthogonales : leur produit d'auto corrélation est nulle. D'un point de vue spectral, (figure 3.17), l'effet de la disparité des erreurs de retard

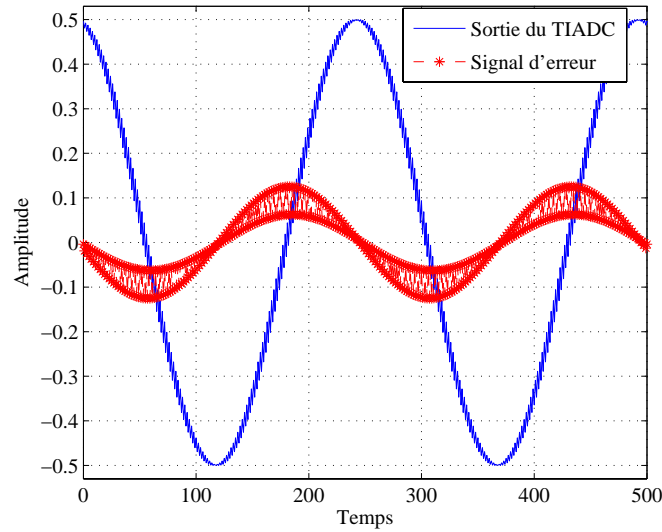


FIG. 3.16 – Erreur du retard d'ouverture dans un TIADC.

à l'ouverture est similaire à celui du gain. Une raie spectrale située à la fréquence $\frac{f_s}{2} - f_0$ vient s'ajouter au spectre de sortie du TIADC. Mais, contrairement à l'erreur de gain, l'amplitude de cette raie augmente avec la fréquence d'entrée. C'est la différence principale entre l'erreur du retard d'ouverture et celle du gain. Cette différence peut être utilisée pour différencier la contribution de l'erreur de gain et de l'erreur de retard à l'ouverture dans le cas d'apparition

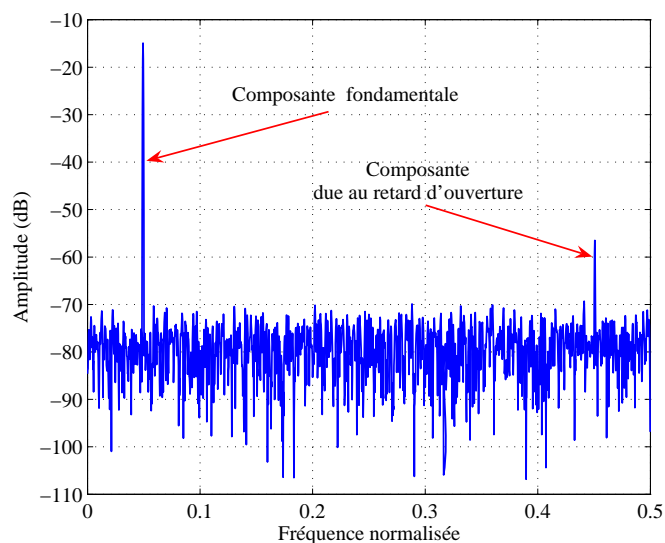


FIG. 3.17 – Effet de la disparité de retard d’ouverture sur le spectre d’un TIADC

d’une raie spectrale à la fréquence $\frac{f_s}{2} - f_0$.

3.4.3 Combinaison des disparités des erreurs

Nous avons vu, dans les parties précédentes, que toute disparité entre les deux convertisseurs entrelacés se traduit dans la plupart des cas par des raies non-harmoniques sur le spectre de sortie du TIADC. Pour des raisons de simplification mais aussi pour mettre l’accent sur l’influence de chaque erreur sur les performances spectrales du système, ces erreurs ont été traitées séparément. Mais, dans le cas réel, deux convertisseurs présentent à la fois, plusieurs caractéristiques différentes. L’objectif de cette partie est de montrer l’influence de la coexistence de toutes les erreurs sur les paramètres spectraux pour en déduire les limites théoriques des TIADC. Nous avons vu que la combinaison de deux erreurs de jitter différentes se traduit par une élévation du plancher du bruit. Nous avons montré par simulations que les disparités des erreurs de non-linéarité peuvent causer des raies non-harmoniques. Pour des raison de complexité d’analyse, les erreurs de non-linéarité ne sont pas traitées. Les erreurs qui seront prises en considération dans cette partie sont l’offset, le gain et le retard d’ouverture.

Supposant que O_1 , G_1 et δt_1 sont respectivement les erreurs d’offset, de gain et de retard d’ouverture du premier convertisseur et O_2 , G_2 et δt_2 sont respectivement les erreurs d’offset, de gain et de retard d’ouverture du deuxième convertisseur. Pour un signal d’entrée $s(t) = A \cos(2\pi f_0 t)$,

la sortie du premier convertisseur est :

$$s_1(n) = G_1 \times A \cos(2\pi f_0 (nT_s + \delta t_1)) + O_1 + eq_1(n) \quad (3.34)$$

Celle du deuxième prend la forme

$$s_2(n) = G_2 \times A \cos(2\pi f_0 (nT_s + \delta t_2)) + O_2 + eq_2(n) \quad (3.35)$$

Soient

$$\begin{aligned} G &= \frac{G_1 + G_2}{2}, & \alpha &= \frac{G_2 - G_1}{2G}, \\ \delta t &= \frac{\delta t_1 + \delta t_2}{2}, & \Delta t &= (\delta t_2 - \delta t_1), \\ O &= \frac{O_1 + O_2}{2}, & \Delta O &= \frac{O_2 - O_1}{2} \end{aligned}$$

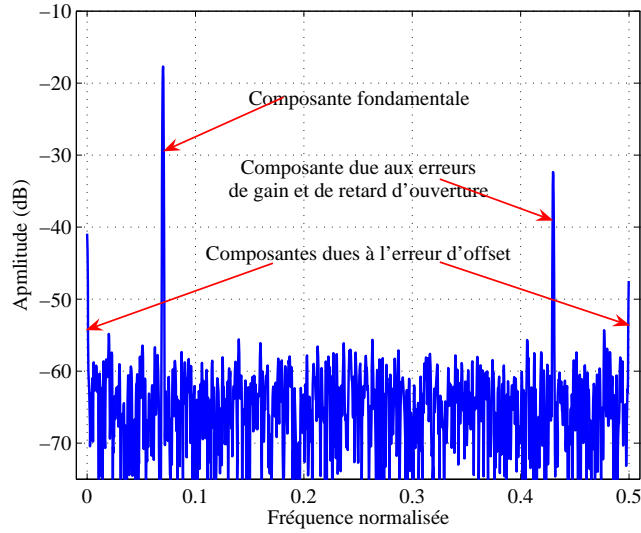


FIG. 3.18 – Effet de la combinaison des erreurs sur le spectre d'un TIADC.

En choisissant la référence des temps à $\delta t = 0$, la sortie du TIADC prenant en compte toutes les erreurs est selon [29] :

$$\begin{aligned} s(n) &= A_s \cos(2\pi f_0 nT_s + \theta_s) + A_n \cos\left(\left(f_0 + \frac{f_s}{2}\right) 2\pi nT_s + \theta_n\right) \\ &+ O + \cos(n\pi) \Delta O + eq(n) \end{aligned} \quad (3.36)$$

où :

- $A_s = AG \sqrt{\cos^2(\pi f_0 \Delta t) + \alpha^2 \sin^2(2\pi f_0 \Delta t)}$
- $A_n = AG \sqrt{\alpha^2 \cos^2(\pi f_0 \Delta t) + \sin^2(2\pi f_0 \Delta t)}$
- $\theta_s = \arctan(\alpha \tan(\pi f_0 \Delta t))$
- $\theta_n = -\arctan\left(\tan\left(\frac{\pi f_0 \Delta t}{\alpha}\right)\right)$

Le spectre de la figure 3.18 montre la coexistence de quatre raies spectrales. La raie fondamentale, une composante DC , une raie à la fréquence $\frac{f_s}{2} - f_0$ due simultanément aux disparités de gain et de retard d'ouverture et une dernière raie à $\frac{f_s}{2}$ due à la disparité des erreurs d'offset. Il est à noter que pour un même convertisseur l'amplitude et la fréquence de ces raies changent avec l'amplitude et la fréquence du signal d'entrée ainsi que les caractéristiques du signal d'horloge appliqué.

3.5 Conclusion

Ce troisième chapitre est fondamental dans notre évolution vers la compréhension de la problématique posée par les TIADC.

Dans un premier temps, nous avons mis l'accent sur l'intérêt des TIADC à travers une étude succincte des architectures de CAN. Nous avons conclu qu'un seul CAN ne peut pas allier simultanément haute résolution, grande rapidité et faible consommation et par conséquent son facteur de mérite ne peut pas répondre aux exigences des nouveaux systèmes de télécommunications mobiles.

Les TIADC sont considérés comme une solution prometteuse pour répondre à ce besoin. Le principal avantage d'une telle structure consiste à l'augmentation linéaire de la fréquence d'échantillonnage avec le nombre de CAN à entrelacer. Cependant, les défauts de fabrication de ces composants se traduisent par des non appariements des erreurs et causent des raies non-harmoniques réduisant ainsi les performances spectrales des TIADC.

Nous avons analysé et formulé mathématiquement les différents effets du non appariements de chaque erreur prise séparément. En effet, nous avons établie un modèle mathématique pour chaque erreur afin d'extraire les dispositions fréquentielles et les amplitudes des raies non-harmoniques résultantes.

Pour valider les modèles mathématiques établies et pour mettre en évidence l'effet de ces composantes parasites sur les performances du TIADC, nous avons extrait les formules du SNR et comparé ces formules à des résultats de simulation d'un convertisseur entrelacé.

Enfin, dans le but de déduire les limites théorique des TIADC, nous avons étudié la combinaison des disparités de toutes les erreurs. Nous avons montré que les différents effets de chaque erreur s'additionnent au niveau de la sortie du TIADC et réduisent sérieusement ses performances.

Étude des méthodes de correction des erreurs de disparités dans les TIADC

Sommaire

4.1	Introduction	71
4.2	Classification des méthodes de correction absolues	72
4.2.1	Classification par emplacement de la correction	72
4.2.2	Classification par mode de correction	75
4.2.3	Comparaison des différentes méthodes de correction absolues	76
4.3	Méthodes en différé pour la calibration des TIADC	77
4.3.1	Méthode utilisant des séquences orthogonales	78
4.3.2	Méthode de calibration par FFT	82
4.4	Méthodes en ligne pour la calibration des TIADC	88
4.4.1	Calibration par choix aléatoire de canal	88
4.4.2	Méthode numérique adaptative	93
4.5	Conclusion	98

4.1 Introduction

Bien que les convertisseurs entrelacés ont progressé en ce qui concerne la fréquence d'échantillonnage, la résolution reste toujours leur point faible. Nous avons analysé les répercussions des différences entre les erreurs caractérisant les CAN d'une structure entrelacée, qui réduisent ses performances spectrales. Augmenter la résolution effective d'un TIADC revient donc à dimi-

nuer l'effet de ces disparités. Toutes ces erreurs sont dues aux imperfections des processus de fabrication. Ces processus (ou procédés) de fabrication sont de plus en plus compliqués et il est quasiment impossible d'avoir les mêmes caractéristiques pour deux convertisseurs ayant les mêmes spécifications. En outre, une fois réalisé, les caractéristiques d'un CAN sont susceptibles de varier avec le vieillissement et les conditions d'utilisation (température notamment). Il est donc nécessaire d'avoir une intervention au cours ou après la fabrication pour remédier à cet effet.

Dans le cadre de la correction des disparités des erreurs dans un TIADC, deux alternatives se présentent :

- la première, appelée correction absolue, consiste à compenser les erreurs de chaque convertisseur. Ainsi, les disparités entre les différents canaux doivent être éliminées. Ces méthodes présentent l'inconvénient d'avoir un coût très élevé.
- la deuxième, appelée correction relative, se base sur la minimisation des disparités des différents CAN. Le principe est de faire tendre les sorties des CAN vers celle d'un convertisseur de référence. Ainsi, les disparités sont réduites.

Dans ce chapitre, nous présentons un état de l'art des différentes méthodes de correction des erreurs d'un CAN dans une première partie. Ensuite, nous détaillons quelques méthodes de calibration d'un TIADC, simulations à l'appui, pour dégager leurs avantages et leurs inconvénients.

4.2 Classification des méthodes de correction absolues

Ici, il s'agit de décrire les méthodes de compensation d'un CAN seul. Ces méthodes peuvent être classées en se basant sur la localisation du processus de correction ou sur le mode de fonctionnement du CAN pendant que ses erreurs sont en phase d'estimation. De ce fait nous parlons de correction analogique et de post-correction numérique d'un côté et de calibration en ligne et en différé de l'autre côté. Nous décrivons dans cette section le principe de ces méthodes.

4.2.1 Classification par emplacement de la correction

4.2.1.1 La correction analogique

Connue sous le nom de pré-distorsion analogique, la correction se fait en appliquant un algorithme de pré-distorsion sur le signal analogique à l'entrée du CAN. Pour un niveau de sortie donné, le signal d'entrée subi un changement pour le rapprocher du signal d'entrée théorique donnant le même niveau de sortie conformément à la figure 4.1. En effet, le convertisseur n'est pas idéal, pour un niveau de sortie L , la plage d'entrée du CAN change d'un convertisseur à un autre par rapport à un convertisseur théorique. Ce sont ces plages d'entrées qui fixent la caracté-

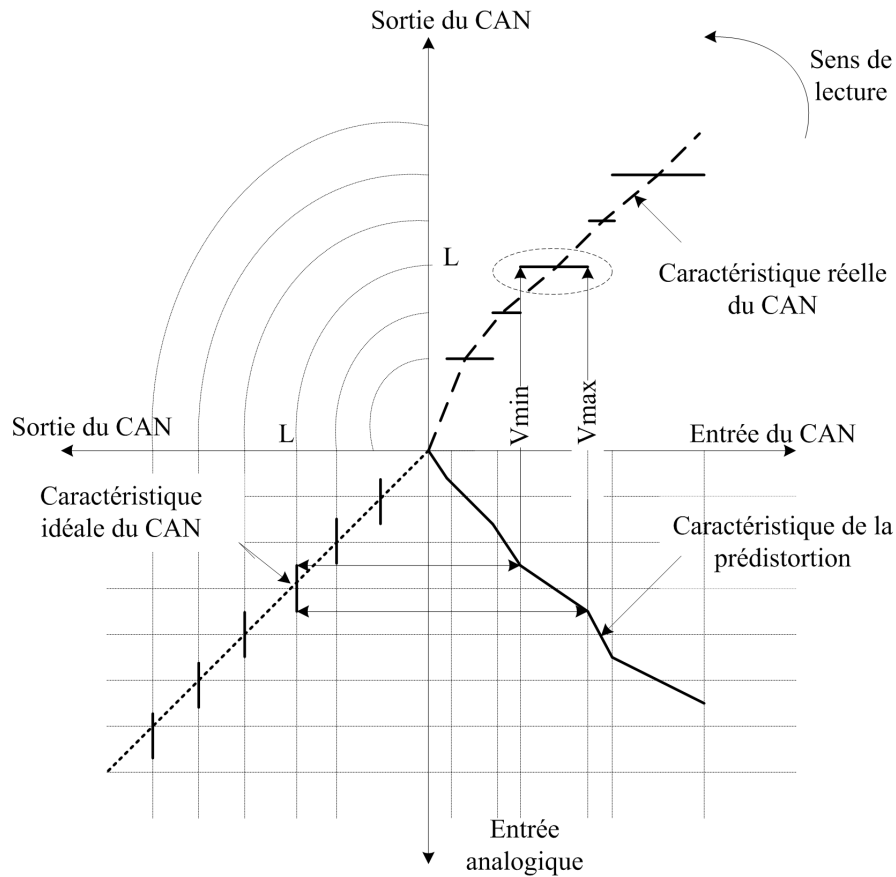


FIG. 4.1 – Le concept de la correction analogique.

ristique de la correction. La fonction de transfert réelle du CAN est présentée dans la figure 4.1. Pour chaque niveau de sortie L est associée une plage de conversion réelle $[V_{min}(L), V_{max}(L)]$. Nous cherchons la plage de conversion théorique, $[V_{th_{min}}(L), V_{th_{max}}(L)]$, sur la caractéristique de transfert théorique donnant le même niveau L à la sortie du convertisseur. Le principe de la méthode consiste à une déformation du signal d'entrée du CAN de manière à faire correspondre, pour chaque niveau L , la plage d'entrée $[V_{th_{min}}(L), V_{th_{max}}(L)]$ à une plage déformée $[V_{min}(L), V_{max}(L)]$ avant d'attaquer le CAN. L'implémentation de cette technique est présentée dans la figure 4.2. La sortie du CNA est utilisée dans le traitement analogique pour calculer la caractéristique de la prédistorsion. Dans la figure 4.2 des délais sont à insérer pour synchroniser les différents blocs.

4.2.1.2 La correction numérique

Le principe de toutes les méthodes de correction numérique est illustré dans la figure 4.3. Pour une plage de conversion située entre $[V_{min}(L), V_{max}(L)]$, nous cherchons le niveau L_{th} sur

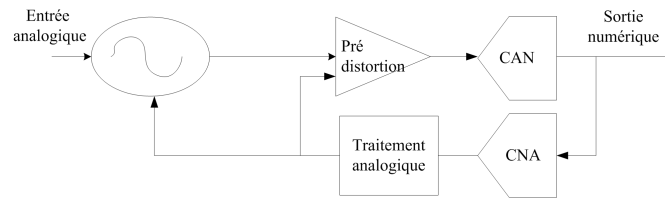


FIG. 4.2 – Implémentation de la correction analogique.

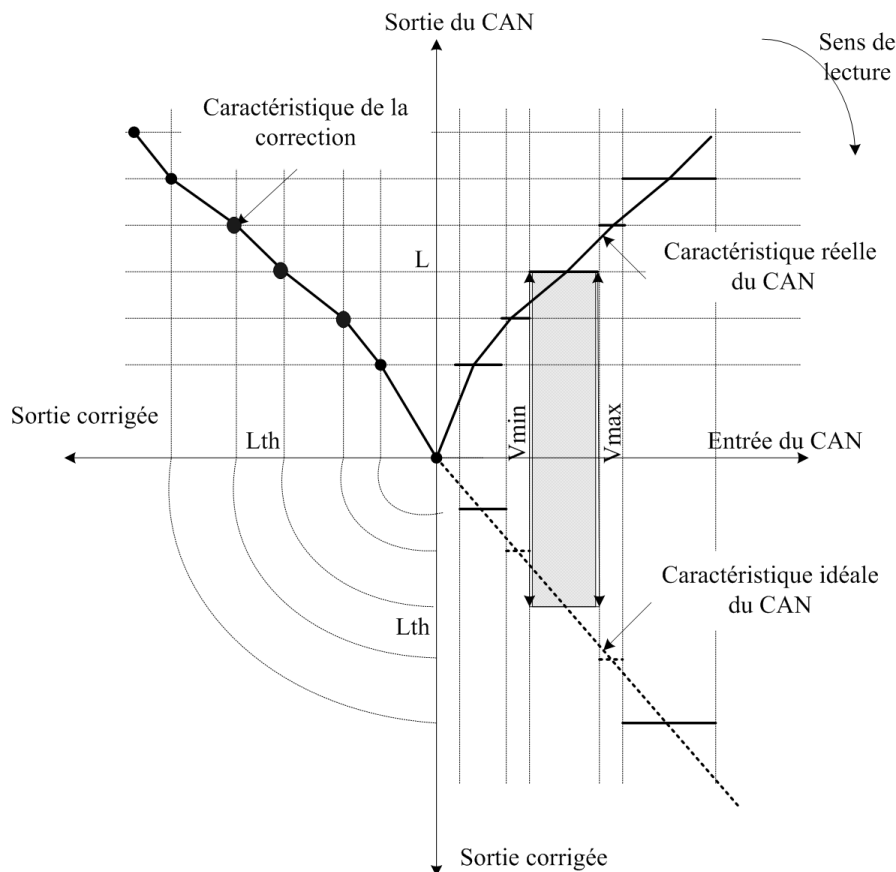


FIG. 4.3 – Le concept de la correction numérique.

la caractéristique de transfert théorique. Ce niveau théorique correspond à l'intersection de la caractéristique de transfert théorique avec le milieu de la plage de conversion. Ainsi, la caractéristique de correction numérique est la courbe qui associe aux niveaux théoriques L_{th} , les niveaux réels L . L'implémentation des méthodes de post-correction dans le cas des CAN se fait dans la partie numérique. Ces techniques de correction profitent donc de l'évolution des technologies de fabrication orientées vers le numérique. Ce sont donc des méthodes de correction plus simples à implémenter. Un traitement numérique de données suffit pour calibrer le CAN.

L'implémentation de ces techniques est présentée dans la figure 4.4. La technique de correction

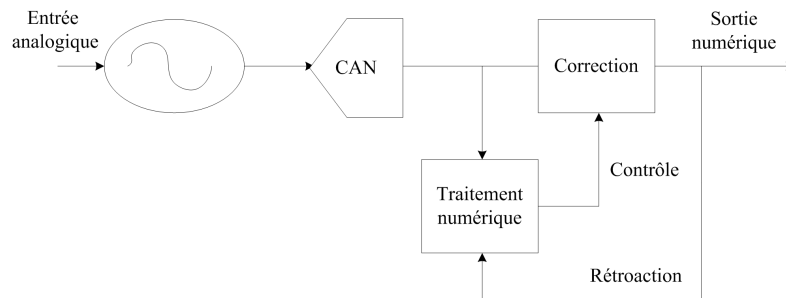


FIG. 4.4 – Implémentation de la correction numérique.

numérique peut être en ligne ou en différé. Le contrôle du bloc de correction peut se faire en topologie directe ou en rétroaction (feedforward ou feedback respectivement). Il est important de signaler que les techniques de corrections numériques profitent de tous les avantages du traitement numérique de l'information ainsi que de la souplesse et de la simplicité de la conception.

4.2.2 Classification par mode de correction

4.2.2.1 La correction en différé

La correction en différé (offline) est un terme donné à tout type de correction où la sortie du composant sous calibration est prise en différé. Dans le cas des CAN ceci veut dire que la correction nécessite une première phase de fonctionnement avant la calibration. Cette phase de diagnostic est menée afin d'enregistrer la sortie du convertisseur avant de le mettre au repos.

Puisqu'il s'agit d'une correction en différé, le CAN doit être excité par un signal précis afin que sa sortie puisse être comparée à la sortie idéale. Ainsi, une correction en différé nécessite un équipement de calibration assez cher pour ne pas introduire des erreurs supplémentaires.

Une des caractéristiques de ces méthodes réside dans la limitation des types d'erreurs concernées par la correction. En effet, pour un signal de test donné et après la phase d'estimation, la caractéristique de correction est calculée. En changeant le signal d'entrée, l'application de la même caractéristique de correction doit éliminer l'ensemble des erreurs caractérisant le CAN. Malheureusement, ceci n'est que partiellement vrai. En effet, les erreurs statiques (erreurs déterministes) sont des caractéristiques intrinsèques du CAN, l'application de la même caractéristique de correction doit corriger ces erreurs quelque soit l'amplitude et la fréquence du signal d'entrée. Mais, les erreurs temporelles, notamment le bruit de phase, dépendent de la fréquence du signal d'entrée. Les méthodes de correction en différé ne sont pas adaptées pour la correction de ce type d'erreurs.

Ce type de calibration ne peut pas être utilisé pour des applications en temps réel car le signal

d'entrée a tendance à changer en amplitude ainsi qu'en fréquence. A chaque changement, un enregistrement de données est nécessaire pour la calibration. Au cours de l'enregistrement et de la calibration, le convertisseur est mis au repos. L'aspect temps réel, n'est donc pas assuré. Par contre, pour des applications de tests et de mesures par exemple, la correction en différé reste une solution efficace. En effet, pour ce type d'applications, on attend la réponse du convertisseur pour un signal d'entrée donné.

4.2.2.2 La correction en ligne

Contrairement à la correction en différé, la correction en ligne peut se faire quand le composant est en ligne (continue son opération). L'intégration matérielle de ces méthodes de correction nécessite souvent une circuiterie supplémentaire. Le coût de ces circuits additionnels est évidemment négligeable devant le coût de l'équipement de test pour une méthode en différé. La sortie du CAN sous calibration peut ne pas être corrigée dès le premier cycle de conversion et pour différentes méthodes de calibration en ligne, la durée de la période d'adaptation est un critère important pour le choix de la méthode de correction à utiliser.

4.2.3 Comparaison des différentes méthodes de correction absolues

Plusieurs interrogations s'imposent dès lors que nous parlons calibration : Où corriger ? Dans quel mode faut-il corriger ? Comment corriger ? Ce sont les questions auxquelles nous avons essayé de répondre tout au long de cette section.

En ce qui concerne le mode de correction, comme nous l'avons expliqué, pour un seul CAN le choix de la méthode en ligne ou en différé dépend de l'application visée. Pour les applications en temps réel c'est la méthode en ligne qui est privilégiée. Pour des applications de mesures, ce sont les méthodes en différé qui s'imposent.

Pour l'emplacement de la correction autour du CAN, il est évident qu'en terme de souplesse et de flexibilité les solutions de correction numérique sont plus avantageuses que les méthodes de correction analogique. En effet, les méthodes numériques profitent de tout l'intérêt des circuits numériques : plus faciles à concevoir, immunité au bruit et aux phénomènes de dérive et reprogrammables. Quoique des méthodes de correction mixte dans le cadre de correction des erreurs de retards d'ouverture pour les TIADC commencent à voir le jour et profitent des avantages de la correction numérique et analogique à la fois, [32].

Le tableau 4.1 présente un résumé des caractéristiques de chaque méthode. Concernant la politique ou la manière à suivre pour mener à bien l'opération de la correction dans la chaîne entrelacée, plusieurs alternatives se présentent. La correction absolue des TIADC consiste à appliquer une méthode de calibration numérique pour chaque convertisseur. A priori c'est la

	Correction analogique	Correction numérique
en différé	Temps de correction élevé. Difficile à concevoir. C'est la technique la moins utilisée.	Coût de correction moins important que celui en analogique. Besoin d'un signal de test de bonne qualité pour la calibration
en ligne	Utilise des blocs de retard analogique. Boucle de rétroaction complexe. Utilise un CNA	Pas de mode de test séparé. Pas besoin d'un générateur de signaux. Utilise des circuits supplémentaires. La technique la moins chère.

TAB. 4.1 – Comparaison des méthodes de correction.

méthode la plus simple puisqu'il s'agit d'appliquer la même méthode sur les différents convertisseurs et elle peut être appliquée sans limitation de nombre de convertisseurs à entrelacer. Mais elle présente deux inconvénients majeurs :

- Le nombre de blocs de calibration augmente linéairement avec le nombre de convertisseurs à entrelacer. Pour une structure comportant 8 convertisseurs par exemple, les blocs de correction sont utilisés 8 fois de la même manière. Par conséquent, la taille et la consommation du circuit final deviennent très importantes.
- Le principe même d'une méthode de calibration, consiste à réduire les erreurs d'un CAN. Il est souvent impossible d'éliminer complètement ces erreurs. Par conséquent, après calibration, les CAN peuvent présenter des caractéristiques de transfert différentes. Le problème de disparité entre les canaux n'est donc pas résolu.

Ainsi, la technique corrigeant chaque voie indépendamment de l'autre ne résout pas totalement le problème de disparité. Notre intérêt s'est donc porté sur les méthodes de correction numériques, en différé et en ligne.

4.3 Méthodes en différé pour la calibration des TIADC

Il s'agit ici d'effectuer un état de l'art des méthodes de correction relatives pour en extraire les avantages et les inconvénients. Cette étude servira de point de départ pour la recherche de nouvelles méthodes. Ici, nous étudions une liste non exhaustive, mais assez représentative, des méthodes de correction existantes dans la littérature. Nous avons incité dans le choix de méthodes à étudier sur la diversité de leur principe, pour cela d'autres méthodes n'ont pas été

étudié comme celles présentées dans [33] et dans [31].

4.3.1 Méthode utilisant des séquences orthogonales

Cette méthode, dont le principe est donné dans [34], se base sur la séparation des canaux par des séquences orthogonales. Ce principe est largement utilisé dans les communications et notamment en CDMA. On montre dans les paragraphes suivants que l'utilisation des séquences orthogonales permet de séparer les contributions des erreurs d'offset et de gain dans une structure entrelacée.

4.3.1.1 Principe

Les erreurs concernées par cette méthode sont les erreurs statiques d'offset et de gain. Le but de la méthode consiste à éliminer l'erreur d'offset et à rapprocher la valeur du gain du convertisseur sous calibration à celle d'un convertisseur de référence. Les convertisseurs utilisés sont donc modélisés par leur erreurs d'offset o_p et de gain g_p . La sortie du convertisseur z_p s'écrit sous la forme :

$$\{z_p\} = o_p + g_p \cdot x + e_q \quad (4.1)$$

où e_q représente le bruit de quantification, supposé très inférieur à l'amplitude du signal d'entrée. Sur ce convertisseur vient s'appliquer un signal aléatoire s comme l'illustre la figure 4.5.

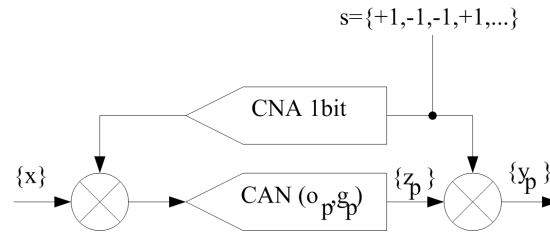


FIG. 4.5 – Illustration de la modulation d'offset.

Le signal s est une séquence de longueur N_t composée d'éléments s_j vérifiant

$$s_j \in \{-1, 1\} \forall j = 1, 2, \dots, N_t \quad (4.2)$$

Avant de multiplier le signal x , la séquence aléatoire s subit une conversion par un CNA 1 bit. La sortie du CNA présente deux niveaux de tension qui sont fixés aux limites de la pleine échelle du CAN. Au niveau circuit, une multiplication de deux signaux analogiques se réalise par des cellules de Gilbert.

L'entrée x ainsi que la sortie du convertisseur z_p sont donc modulées par la séquence s . La sortie y_p de la figure 4.5 est alors :

$$y_p = (o_p + g_p \cdot x \bullet s) \bullet s = o_p \bullet s + g_p \cdot x \quad (4.3)$$

L'opérateur \bullet représente l'opération de la modulation (équivalent à \times dans les figures). L'égalité dans l'équation 4.3 est toujours vraie puisque $\{s\} \bullet \{s\} = 1$ (propriété de l'orthogonalité). En d'autres termes la modulation de l'entrée et de la sortie du convertisseur par une séquence $\{s\}$ revient à la modulation de l'erreur d'offset par la même séquence $\{s\}$. C'est cet effet qui servira à la correction des erreurs statiques dans un TIADC conformément à la figure 4.6.

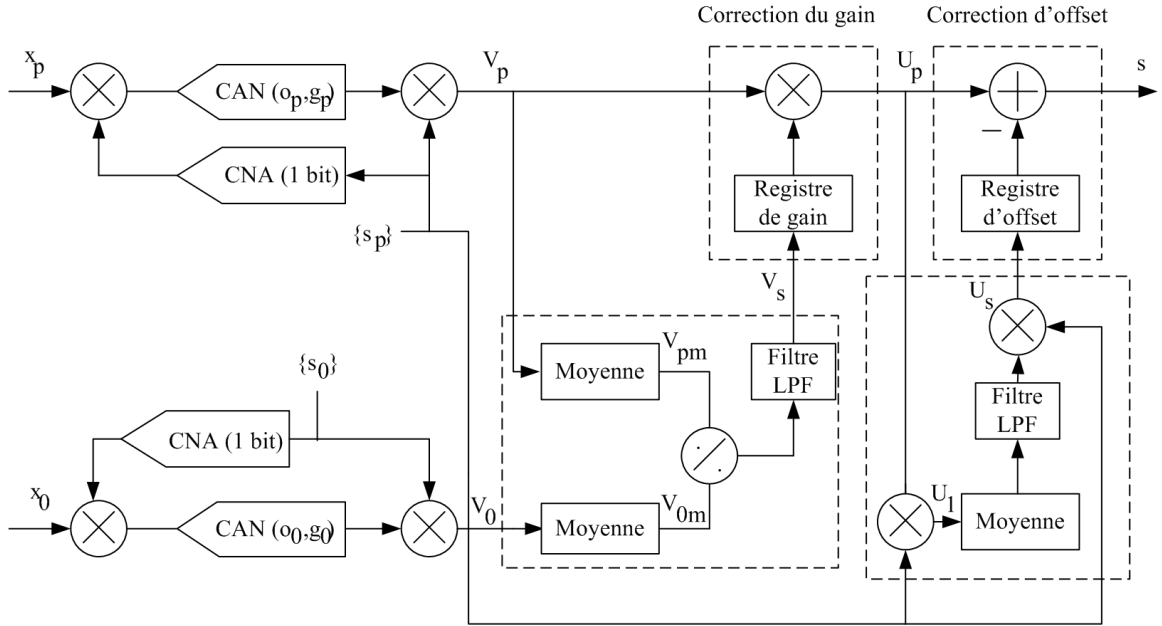


FIG. 4.6 – Synoptique de la calibration d'un TIADC utilisant les séquences orthogonales.

Le même principe de la modulation d'offset est appliqué pour séparer les contributions d'offset et de gain. Deux séquences orthogonales (s_1 et s_2) de tailles N_t sont alors utilisées satisfaisants la relation 4.2 et les relations suivantes :

$$\sum_{j=1}^{N_t} s_{1j} = 0 \quad (4.4)$$

$$\sum_{j=1}^{N_t} s_{2j} = 0 \quad (4.5)$$

$$\sum_{j=1}^{N_t} s_{1j}s_{2j} = 0 \quad (4.6)$$

La figure 4.6 explique schématiquement comment les séquences orthogonales peuvent être utilisées pour calibrer une voie p du TIADC, modélisée par le couple d'erreur (o_p, g_p) , utilisant un convertisseur de référence modélisé par le couple d'erreur (o_0, g_0) . Le circuit numérique de la correction d'une voie utilise trois filtres d'ordre N_t représentés dans la figure 4.6 par les blocs

'Moyenne' ($\sum_{N_t} \equiv \frac{1}{N} \sum_{l=0}^{N-1} z^{-l}$). Ces trois filtres récupèrent des informations permettant l'estimation de l'erreur d'offset et de gain. Les filtres passe-bas permettent de lisser les informations à leurs entrées pour avoir une meilleure précision des valeurs d'erreurs estimées.

La correction se fait donc en deux phases : une première phase d'estimation des erreurs de gain et d'offset et une deuxième pour la correction proprement dite. Pour calibrer une voie de la chaîne entrelacée, l'offset de chaque convertisseur est complètement éliminé alors que le gain de chaque voie est fixé à celui du convertisseur de référence. Ainsi, tous les convertisseurs présentent le même gain et un offset égale à zéro. Exposons maintenant le principe théorique.

Supposant qu'il ne s'agit que de deux voies à calibrer. L'application de la moyenne sur le signal de la voie p donne :

$$V_{pm} = \sum_{j=1}^{N_t} (o_p \bullet s_{pj} + g_p x_p) = g_p x_p \quad (4.7)$$

De la même manière $V_{0m} = g_0 x_0$. Ainsi, après division de V_{0m} par V_{pm} , le signal V_s est le rapport des deux valeurs de gain, soit $\frac{g_0}{g_p}$. Cette sortie est sauvegardée dans un registre de gain pour multiplier le signal v_p . Le signal U_p s'écrit donc sous la forme :

$$U_p = \frac{g_0}{g_p} (o_p \bullet s_p + g_p x_p) = o_p \frac{g_0}{g_p} \bullet s_p + g_0 x_p \quad (4.8)$$

Ensuite, la séquence aléatoire s_p multiplie le signal U_p pour donner U_1

$$U_1 = s_p \times \left[o_p \frac{g_0}{g_p} \bullet s_p + g_0 x_p \right] = o_p \frac{g_0}{g_p} + g_0 s_p x_p \quad (4.9)$$

Le signal U_1 est moyenné et ensuite multiplié par la même séquence orthogonale s_p pour donner U_2 qui vaut $o_p \frac{g_0}{g_p} s_p$. Enfin, la soustraction du signal U_p (de l'équation 4.8) avec le signal U_2 donne le signal de sortie de la calibration s qui s'écrit sous la forme :

$$s = g_0 x_p \quad (4.10)$$

La voie p est donc calibrée : elle présente un offset égale à zéro et un gain égal à celui de la voie de référence, g_0 . Toutes les autres voies à calibrer présenteront les mêmes valeurs d'erreurs. Les disparités d'offset et de gain entre les différentes voies du *TIADC* sont éliminées.

4.3.1.2 Simulations

Pour valider cette méthode, nous avons effectué des simulations avec le modèle du *TIADC* détaillé dans [35]. Il s'agit d'un modèle comportementale développé sous Simulink. Les paramètres d'erreurs utilisés sont : $O_1 = 1.10^{-2}$ V, $O_2 = 5.10^{-2}$ V, $G_1 = 1 + 2.10^{-2}$ et $G_2 = 1 + 5.10^{-2}$. Les erreurs de retard d'ouverture sont fixés à zéro. La résolution de chaque CAN est de 8 bits et la condition d'échantillonnage cohérent est respectée. La correction est réalisée par des fonctions

MATLAB [36].

Pour calibrer un TIADC à deux canaux, la méthode de calibration, telle qu'elle est proposée dans [34], emploie un convertisseur supplémentaire modélisé par son couple d'erreur (O_0, G_0) . Les séquences orthogonales générées relatives aux voies 0, 1 et 2 sont des séquences répétitives à base de $s_0 = [1, 1, -1, -1]$, $s_1 = [1, -1, 1, -1]$ et $s_2 = [1, -1, -1, 1]$. Ces trois séquences vérifient les propriétés de l'orthogonalité des équations (4.4), (4.5) et (4.6). Les figure 4.7 et 4.8 montrent les spectres de sorties avant et après la correction en respectant la procédure de calibration illustrée dans la section précédente. Après la correction le SNR augmente de 10 dB alors que le $SFDR$ passe de 31 dB à 62 dB.

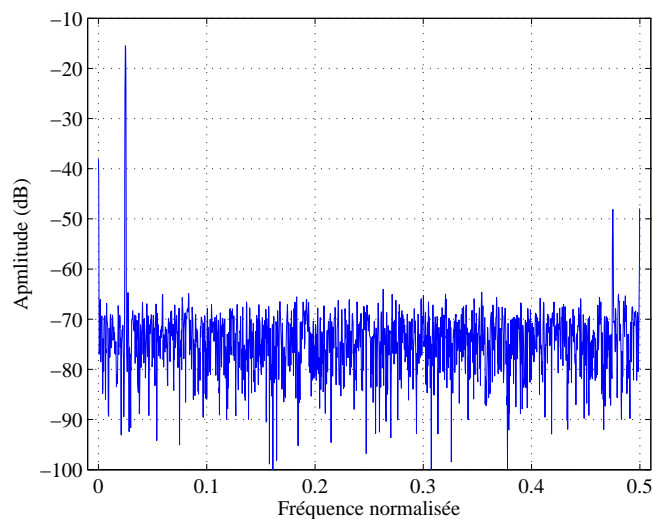


FIG. 4.7 – Spectre de sortie du TIADC avant correction.

4.3.1.3 Discussions

La méthode proposée est une méthode en différé numérique. Les erreurs traitées sont les erreurs statiques d'offset et de gain. Théoriquement, il n'y a pas de limitation par rapport au nombre de convertisseurs à calibrer. En effet, des blocs de multiplexage et de démultiplexage sont utilisés pour étendre l'application de la méthode sur un nombre de CAN supérieur à 2. Ce point constitue un avantage par rapport à d'autres méthodes que nous citerons dans la suite.

Un deuxième avantage est lié au coût constant de la correction. En effet, quelque soit le nombre de CAN à entrelacer, la méthode en question présente quasiment le même coût. Toutes les opérations nécessaires à la calibration sont principalement autour du convertisseur de référence.

Néanmoins, la méthode de calibration peut être optimisée en incluant le convertisseur de référence dans la structure du TIADC. L'utilisation d'un CAN supplémentaire comme référence

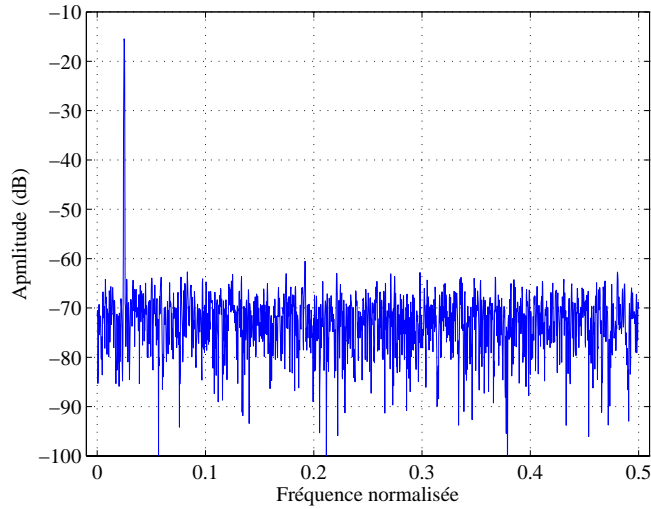


FIG. 4.8 – Spectre de sortie du TIADC après correction.

constitue un surdimensionnement du TIADC. Il est à noter aussi, que la méthode utilise une circuiterie complexe gourmande en terme de consommation et d'occupation de surface silicium : deux générateurs de séquences orthogonales, des filtres *LPF*, des filtres pour extraire la moyenne et deux CNA.

4.3.2 Méthode de calibration par FFT

Cette méthode, proposée dans [37], est une méthode logicielle basée sur la transformation de Fourier rapide (FFT, *Fast Fourier Transform*) afin d'estimer les valeurs de gain et d'offset de chaque CAN dans la structure entrelacée. Cela permet, ensuite, une compensation.

4.3.2.1 Principe

Il s'agit d'une méthode mathématique se basant sur l'estimation des contributions spectrales des erreurs d'offset et de gain. Le TIADC est composé de M convertisseurs, chacun d'entre eux est modélisé par un échantillonneur et un quantificateur. Un multiplexeur est utilisé pour entrelacer les données de sorties des quantificateurs. Chaque convertisseur est caractérisé par son propre couple d'erreur (O_i, G_i). Les instants d'échantillonnage pour chaque convertisseur sont donnés par :

$$T_{s,i}[n] = \frac{1}{f_s} \left(n + \frac{i-1}{M} \right) + t_0 \quad i = 1, 2, \dots, M \quad (4.11)$$

n est un entier représentant l'ordre de l'échantillon, i est l'indice du convertisseur et t_0 représente l'instant initial d'échantillonnage qui peut être forcé à zéro. Le signal à la sortie de chaque convertisseur s'écrit sous la forme :

$$x_i[n] = x(t)_{t=T_{s,i}[n]} = (1 + G_i)x(T_{s,i}[n]) + O_i + e_q(n) \quad (4.12)$$

Considérons un signal d'entrée sinusoïdal donné par :

$$x(t) = A \cos(2\pi f_0 t + \varphi) \quad (A \gg e_q) \quad (4.13)$$

Le signal d'erreur de chaque convertisseur est :

$$e_i(t) = (G_i A \cos(2\pi f_0 t + \varphi) + O_i + e_q(t)) \times \sum_{n=-\infty}^{+\infty} \left(\delta \left(t - \left(n + \frac{i-1}{M} \right) T_s \right) \right) \quad (4.14)$$

En négligeant l'effet de quantification et en se plaçant dans les conditions d'échantillonnage cohérent, le signal d'erreur de la structure entrelacée est une fonction discrète et périodique. Dans ces conditions, les DFT (Transformé de Fourier Discrète) des erreurs d'offset et de gain sont données par les relations suivantes :

$$E_{OFF} [i] = \sum_{k=1}^M O_k \exp(-j(2\pi/M)(i-1)(k-1)) \quad (4.15)$$

et

$$E_{GAIN} [i] = \sum_{k=1}^M G_k A \cos(2\pi f_0 T_{s,k} [n] + \varphi) \exp(-j(2\pi/M)(i-1)(k-1)) \quad (4.16)$$

Afin de calculer les valeurs des offsets et des gains, la représentation matricielle des équations 4.15 et 4.16 est :

$$\begin{pmatrix} E_{OFF} (f'_1) \\ E_{OFF} (f'_2) \\ \vdots \\ E_{OFF} (f'_M) \end{pmatrix} = k_1 \begin{pmatrix} (W_M^0)^0 & (W_M^1)^0 & \cdots & (W_M^{M-1})^0 \\ (W_M^0)^1 & (W_M^1)^1 & \cdots & (W_M^{M-1})^1 \\ \vdots & \vdots & \ddots & \vdots \\ (W_M^0)^{M-1} & (W_M^1)^{M-1} & \cdots & (W_M^{M-1})^{M-1} \end{pmatrix} \begin{pmatrix} O_1 \\ O_2 \\ \vdots \\ O_M \end{pmatrix} \quad (4.17)$$

eth

$$\begin{pmatrix} E_{GAIN} (f''_1) \\ E_{GAIN} (f''_2) \\ \vdots \\ E_{GAIN} (f''_M) \end{pmatrix} = k_2 \begin{pmatrix} (W_M^0)^0 & (W_M^1)^0 & \cdots & (W_M^{M-1})^0 \\ (W_M^0)^1 & (W_M^1)^1 & \cdots & (W_M^{M-1})^1 \\ \vdots & \vdots & \ddots & \vdots \\ (W_M^0)^{M-1} & (W_M^1)^{M-1} & \cdots & (W_M^{M-1})^{M-1} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ G_M \end{pmatrix} \quad (4.18)$$

Où $k_1 = \frac{1}{T}$, $k_2 = \left(\frac{A}{2T}\right) \exp^{j\varphi}$, $W_M = \exp^{-j\left(\frac{2\pi}{M}\right)}$ et f' et f'' sont les fréquences des composantes spectrales des erreurs d'offset et de gain respectivement.

Finalement, en utilisant les équations 4.17 et 4.18, les vecteurs d'offset et de gain pour un TIADC sont :

$$[O] = \frac{1}{K_1} [W]^{-1} [E_{OFF}] \quad (4.19)$$

$$[G] = \frac{1}{K_2} [W]^{-1} [E_{GAIN}] \quad (4.20)$$

Où W est une matrice symétrique dans les éléments sont :

$$W_{lc} = W_{cl} = (W_M^{c-1})^{l-1} \quad (4.21)$$

l et c représentent les lignes et les colonnes de la matrice W .

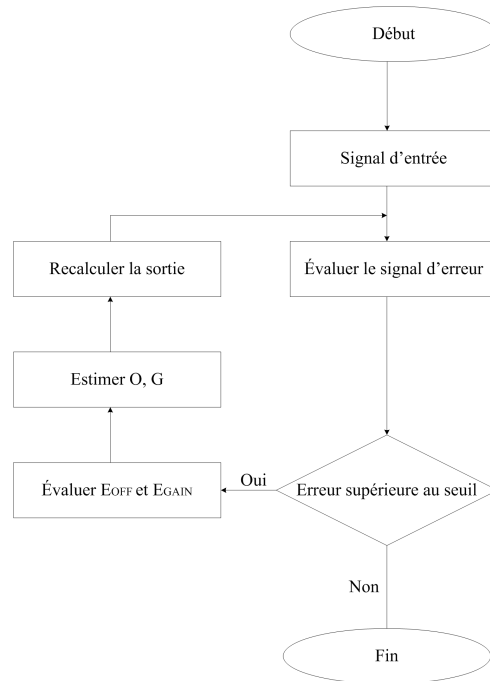


FIG. 4.9 – Procédure de calibration.

Les équations 4.15-4.20 montrent que la détermination des erreurs d'offset et de gain dans une structure entrelacée est un système de $2M$ équations à $2M$ coefficients inconnues. La procédure de calibration utilisée, présentée à la figure 4.9, est résumée dans les étapes suivantes :

1. Appliquer un signal d'entrée suffisamment précis (filtré) en respectant la condition de l'échantillonnage cohérent.
2. Évaluer le signal d'erreur par FFT.
3. Si le maximum de l'amplitude du signal d'erreur est inférieur au bruit de quantification plus 3 dB, la procédure est terminée. Sinon, exécution de l'étape 4.
4. Sélection des composantes spectrales associées aux erreurs d'offset et de gain pour chaque convertisseur.
5. Utiliser les équations 4.19 et 4.20 pour évaluer l'erreur de gain et d'offset pour chaque convertisseur.
6. Estimer la sortie compensée pour chaque convertisseur conformément à la relation suivante :

$$\hat{y}_i[n] = \frac{y_i[n] - O_i}{G_i} \quad i = 1, 2, \dots, M \quad (4.22)$$

7. Mettre à jour la sortie :

$$\hat{y}_i[n] = y_i[n] \quad i = 1, 2, \dots, M \quad (4.23)$$

8. Exécuter l'étape 2 si le nombre maximum d'itération n'est pas dépassé.

Il est à noter que la méthode proposée est itérative dans le cas où le TIADC à calibrer comprend des erreurs de retard d'ouverture. Par conséquent, deux constantes sont utilisées : un nombre d'itération maximal (non précisé dans [37]) et un seuil pour évaluer le signal d'erreur. Le seuil est choisi à 3 dB au dessus du bruit de quantification.

4.3.2.2 Simulations

Le modèle du TIADC utilisé pour les simulations est celui détaillé dans le paragraphe précédent en se basant sur une approche de description matérielle avec VHDL-AMS comme langage de description [38]. Les paramètres d'erreurs utilisés sont : $O_1 = 1.10^{-2}$ V, $O_2 = 5.10^{-2}$ V, $G_1 = 1 + 2.10^{-2}$ et $G_2 = 1 + 5.10^{-2}$. Les erreurs de retard d'ouverture sont fixés à zéro. La résolution des convertisseurs est égale à 8 bits. La condition d'échantillonnage cohérent est aussi respectée. La correction proprement dite, est réalisée par des fonctions MATLAB [36]. Les figures 4.10 et 4.11 montrent les spectres de sorties avant et après la correction en respectant la procédure de calibration illustrée dans la section précédente et en effectuant une seule itération car les erreurs de retard à l'ouverture sont fixées à zéro. La méthode proposée permet d'éliminer les raies spectrales dues aux disparités des erreurs d'offset et de gain.

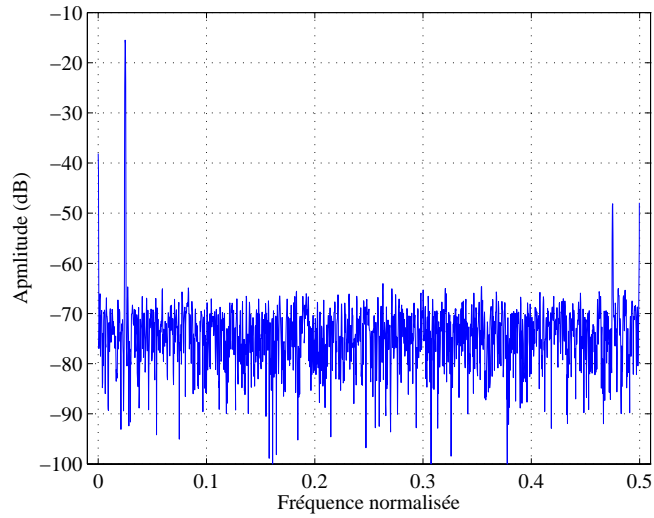


FIG. 4.10 – Spectre de sortie avant la correction.

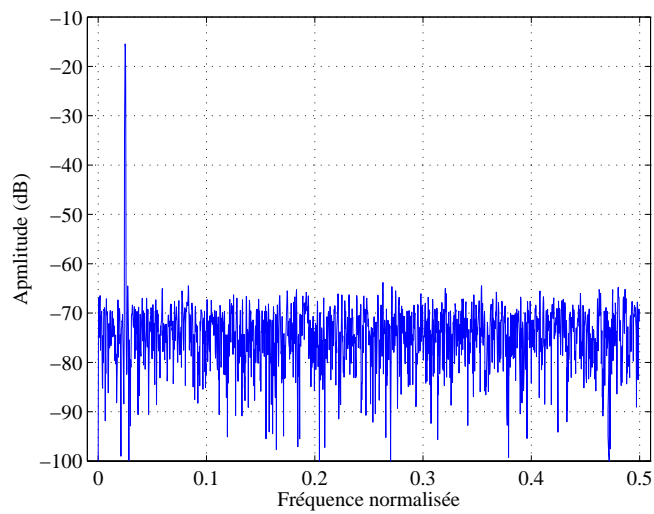


FIG. 4.11 – Spectre de sortie après la correction.

4.3.2.3 Discussions

Il s'agit d'une méthode numérique en différé. Pour faire la correction, il faut que la sortie du TIADC soit disponible. L'application de la méthode ne limite pas le nombre de convertisseurs utilisés dans l'entrelacement, c'est un avantage de la méthode.

Un système de $2M$ équations à $2M$ inconnus est nécessaire pour faire la calibration. L'utilisation de cette méthode nécessite aussi la connaissance du signal d'entrée et du signal d'horloge. Ceci

est indispensable pour le calcul des fréquences de disparités des erreurs d'offset et de gain.

Néanmoins, cette méthode présente un inconvénient majeur lié au nombre d'itérations nécessaires pour la calibration. Deux cas de figures se présentent :

- le TIADC ne présentant que des erreurs d'offset et de gain : une seule itération est donc nécessaire. Durant cette itération, le signal d'erreur est évalué, les valeurs de E_{OFF} et E_{GAIN} sont déterminées, les valeurs de O et de G correspondant aux erreurs d'offset et de gain sont extraites et la sortie sans erreur est recalculée. La puissance du signal d'erreur est inférieure au seuil et la calibration est réalisée.
- le TIADC présente des erreurs d'offset, de gain mais aussi des erreurs de retard d'ouverture. La méthode de calibration est adaptée uniquement à la correction d'offset et de gain. Ainsi, à la fin de la première itération, la puissance de l'erreur est toujours supérieure au seuil fixé à 3 dB au dessus de la puissance du bruit de quantification. La figure 4.12 montre que pour remédier à ce fait, plusieurs itérations sont alors nécessaires. Les valeurs d'offset et de gain changent pour chaque itération afin de tenir compte des erreurs de retard à l'ouverture. Par conséquent, après une phase d'adaptation, la valeur du $SFDR$ augmente linéairement avec le nombre d'itérations. Pour les paramètres d'erreurs fixés dans les sections précédentes, une différence de retard d'ouverture de $0.04 T_s$ entre les deux CAN exige 24 itérations pour éliminer complètement la raie localisée à $\frac{f_s}{2} - f_0$. Le nombre d'itérations peut augmenter dans des proportions considérables avec les paramètres d'erreurs utilisés avec un risque de convergence vers des minimum locaux, ou de divergence.

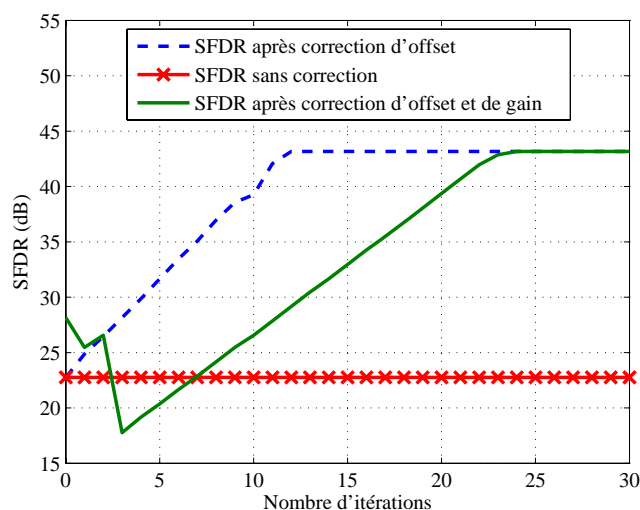


FIG. 4.12 – Évolution du SFDR.

En conclusion, pour tenir compte des erreurs de retard d'ouverture, le nombre d'itérations requis est important et par conséquent le temps nécessaire pour la calibration est élevé. Dans ces conditions, l'implémentation matérielle est très coûteuse.

4.4 Méthodes en ligne pour la calibration des TIADC

4.4.1 Calibration par choix aléatoire de canal

4.4.1.1 Principe

Cette méthode a été proposée dans [39] et reprise dans [28]. Elle se base sur la génération de nombres aléatoires pour la sélection du convertisseur actif du TIADC. En effet, il est vrai que les raies spectrales de disparités des erreurs sont dues aux erreurs dans le processus de fabrication des CAN. Mais, la façon de récupérer les données est aussi responsable du phénomène d'apparition des raies parasites. Les données sont toujours prises en mode ping-pong dans le cas de deux CAN à entrelacer, et d'une façon générale, pour M convertisseurs, les données sont récupérées d'une manière circulaire, introduisant ainsi une périodicité sur l'erreur et donc l'apparition des raies non harmonique.

Afin de remédier aux raies de disparités, une idée consiste à sélectionner d'une façon aléatoire le CAN actif du TIADC. Il s'agit de générer une séquence aléatoire pour la sélection du convertisseur, mais comme l'effet parfaitement aléatoire n'existe pas, nous parlons donc de séquence pseudo aléatoire dans la suite.

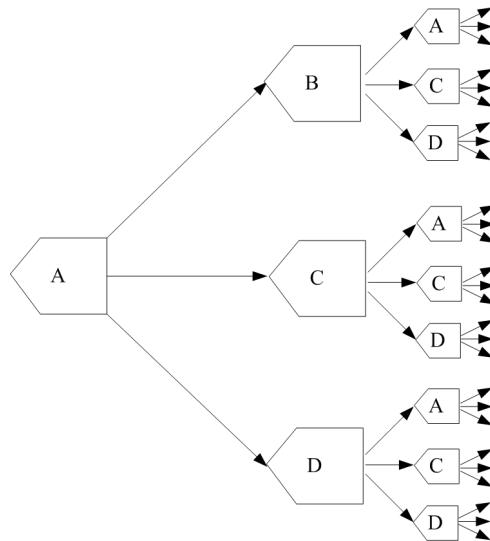


FIG. 4.13 – Configuration par choix de deux convertisseurs.

Dans le cas de quatre CAN à entrelacer, la séquence conventionnelle $A \leftrightarrow B \leftrightarrow C \leftrightarrow D \leftrightarrow$

$A \leftrightarrow B \leftrightarrow C \dots$ laisse place à plusieurs configurations possibles. La figure 4.13 illustre la première configuration. Le principe découle du fait qu'un convertisseur ne peut pas échantillonner les données à double vitesse. Ainsi, une fois le convertisseur A choisi, le prochain convertisseur à sélectionner ne peut pas être celui-ci. Sinon, il faut attendre le prochain front montant de l'horloge du convertisseur A . Dans cette situation la fréquence d'échantillonnage du TIADC est égale à celle du convertisseur A . Donc, après le choix du convertisseur A , trois possibilités sont possibles : B , C ou D . Si B est sélectionné, il faudra choisir un convertisseur dans l'ensemble (A, C, D) et ainsi de suite...

L'entrelacement de M convertisseurs conformément à cette configuration, ne permet que de doubler le taux d'échantillonnage du système. En effet le convertisseur choisi au cycle n , ne peut pas l'être au cycle $n + 1$ mais peut être sélectionné au cycle $n + 2$. Ainsi, une période d'échantillonnage T_s de la structure entrelacée, ne présente pas M échantillons mais désormais 2. Une autre configuration présentée dans la figure 4.14 permet une meilleure optimisation de la fréquence d'échantillonnage du TIADC. La décision du choix du convertisseur se fait par élimination par rapport à celui qui a été choisi lors des deux instants d'échantillonnage précédents. Ainsi, la fréquence d'échantillonnage est triplée. D'une façon générale, pour une structure entrelacée contenant M convertisseurs, la fréquence d'échantillonnage du système est au meilleur des cas égale à $(M - 1) f_s$. Ceci constitue un réel inconvénient de la méthode.

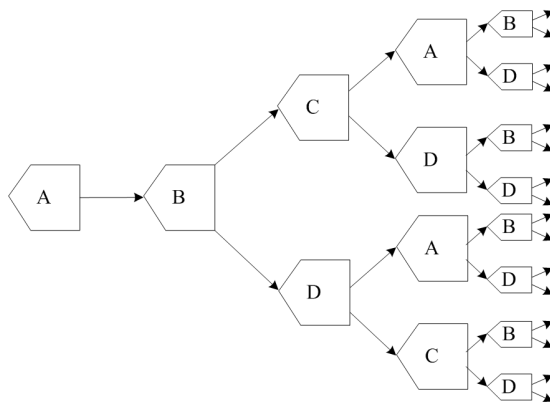


FIG. 4.14 – Configuration par choix de trois convertisseurs.

4.4.1.2 Simulations

Les simulations de cette partie ont pour but de valider la méthode et de montrer que le $SFDR$ diminue avec le choix aléatoire des canaux. Le modèle VHDL-AMS du TIADC utilisé pour ces simulations est celui détaillé en annexe B. La génération des nombres pseudo-aléatoires est montrée en annexe C. Dans cette partie nous tenons compte des configurations illustrées

dans les figures 4.13 et 4.14. Pour faire une étude complète de la structure, nous utilisons une troisième configuration dans la quelle la sélection du canal se fait directement avec le nombre pseudo-aléatoire généré. La structure du TIADC est représenté dans la figure 4.15.

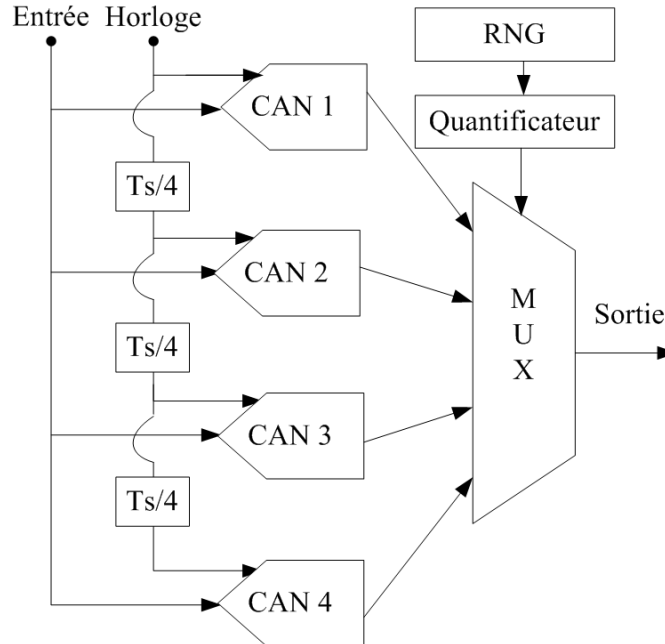


FIG. 4.15 – Structure du TIADC utilisant une correction par génération de nombres aléatoires.

Le TIADC est modélisé de sorte que les vecteurs d'erreurs soient :

- erreurs d'offset : $\{0.02, 0.05, 0.03, 0.08\}$ V ;
- erreurs de gain : $\{1.02, 1.08, 1.13, 0.97\}$;
- erreurs de retard à l'ouverture : $\{0.02, 0.09, 0.045, -0.07\} T_s$.

L'amplitude du signal d'entrée est fixée à 1 V et la résolution de chaque convertisseur est de 8 bits. Les signaux d'horloges sont conçus de manière à assurer l'entrelacement temporel. Les résultats de simulation de la structure représentée par la figure 4.15 sont reportés sur le spectre de la figure 4.16.

Les raies spectrales de disparités des erreurs sont conformes à la théorie. En effet, le spectre de la figure 4.16 montre la coexistence de trois raies de disparité d'offset : une composante *DC*, une raie à $\frac{f_s}{2}$ et une autre à $\frac{f_s}{4}$ et de trois raies de disparités de gain et de retard d'ouverture localisées à $\frac{f_s}{4} - f_0$, $\frac{f_s}{4} + f_0$ et $\frac{f_s}{2} - f_0$. Il est évident que ces raies diminuent la valeur du *SFDR* de la structure entrelacée.

Maintenant, nous supposons que les convertisseurs sont sélectionnés d'une manière aléatoire à l'aide du générateur de nombres pseudo aléatoires. Pour mettre l'accent sur l'effet de la sélection aléatoire des CAN, nous nous tenons pas compte dans un premier temps des configurations

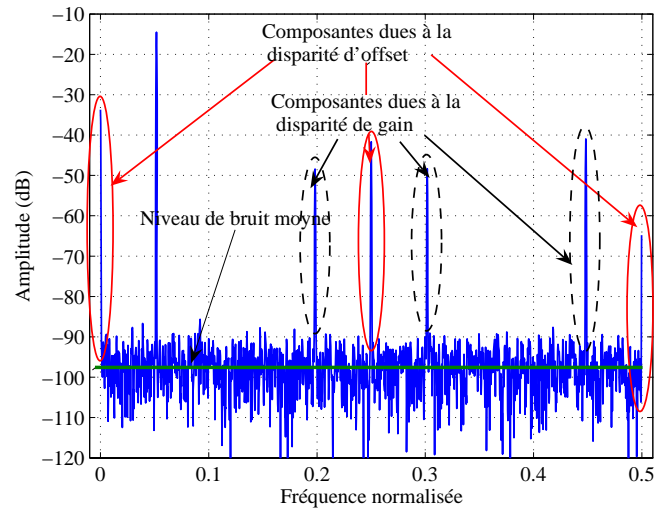


FIG. 4.16 – Spectre de sortie d'un TIADC à quatre CAN.

présentées dans les figures 4.13 et 4.14 : le convertisseur A par exemple peut être sélectionné à deux instants d'échantillonnage successifs. Les résultats sont présentés dans la figure 4.17.

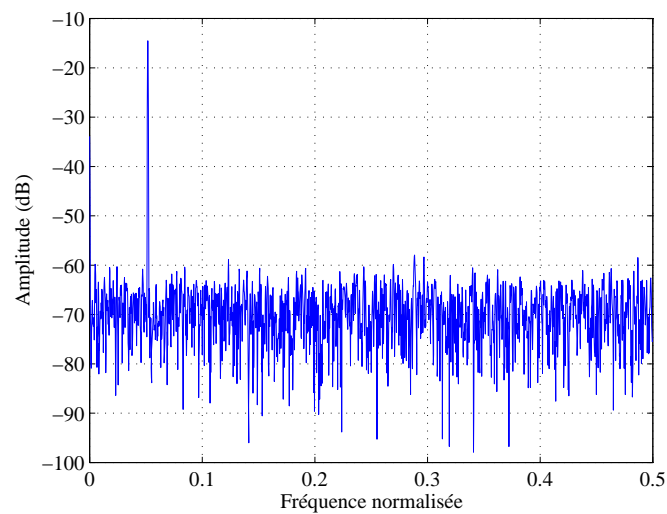


FIG. 4.17 – Spectre de sortie après correction par choix aléatoire de canal.

Les raies de disparité sont complètement éliminées (à l'exception de la composante DC) et le $SFDR$ augmente de 7 dB. Cependant, le niveau moyen de bruit augmente considérablement (plus de 25 dB). Cette variation est due à la sélection aléatoire des canaux. En effet, le phénomène de dithering élimine la périodicité sur les erreurs d'offset, de gain et d'ouverture, mais, ces erreurs existent encore : elles se manifestent sous la forme d'un bruit aléatoire additif au bruit

de quantification ce qui explique l'augmentation du plancher du bruit.

En tenant compte des configurations illustrées dans les figures 4.13 et 4.14 le taux d'échantillonnage du TIADC est respectivement le double et le triple de celui d'un seul CAN. Cependant, les spectres des figures 4.18 et 4.19, relatifs aux configurations des figures 4.13 et 4.14, montrent une élévation du plancher du bruit par rapport au spectre de la figure 4.17. Cette élévation du plancher de bruit vaut 2.87 dB pour la configuration de la figure 4.13 et 4.43 dB pour la configuration de la figure 4.14. Encore une fois, la précision et la rapidité ne progressent pas ensemble.

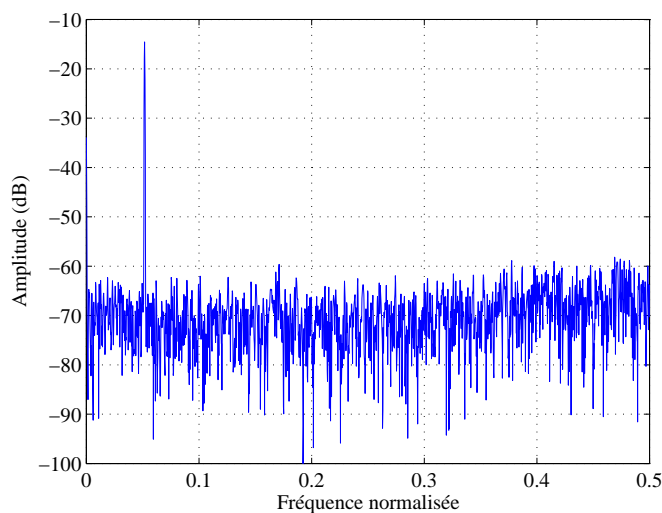


FIG. 4.18 – Spectre de sortie après correction utilisant la première configuration.

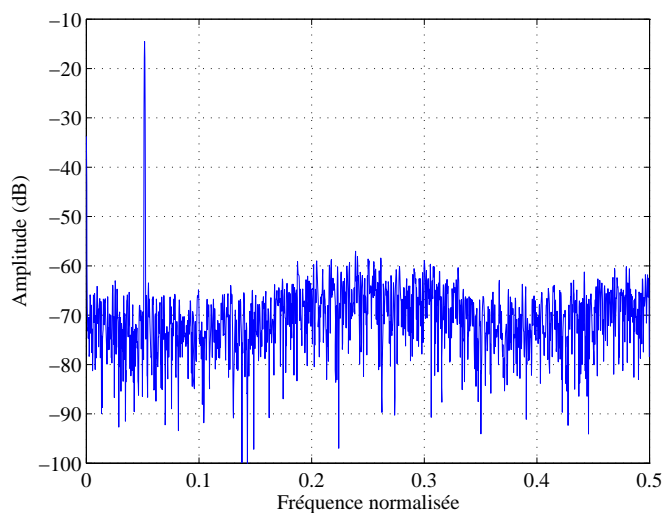


FIG. 4.19 – Spectre de sortie après correction utilisant la deuxième configuration.

La meilleure précision coïncide avec la rapidité la plus basse. Certes, la sélection aléatoire des CAN du TIADC élimine complètement les raies des disparités et augmente par la suite la valeur du $SFDR$. Mais, l'élévation du plancher du bruit pour assurer une rapidité maximale diminue le SNR du TIADC.

4.4.1.3 Discussions

La méthode proposée dans [39] est une méthode en ligne numérique. Elle se base sur le principe de sélection aléatoire des CAN d'un TIADC contrairement à la manière circulaire. L'implémentation matérielle d'un générateur de nombre pseudo-aléatoire est assez simple. Nous avons montré par simulations que la sélection aléatoire des canaux de sortie d'un TIADC permet d'éliminer les raies de disparités des différentes erreurs. Il s'agit donc d'une méthode de correction très intéressante pour des TIADC utilisés dans des applications de télécommunication, notamment dans la détection de porteuse pour les applications Radar où la valeur du $SFDR$ est prépondérante. Cependant, la sélection aléatoire des CAN pose certains problèmes. En effet, pour M convertisseurs à entrelacer, la fréquence d'échantillonnage ne peut, en aucun cas, atteindre Mf_s . La configuration du générateur de nombres pseudo-aléatoires qui assure une fréquence d'échantillonnage égale à $(M - 1)f_s$ devient plus complexe. En effet, il faut utiliser des mémoires pour tenir compte des dernières $(M - 2)$ valeurs avant de générer la valeur du CAN actif.

Finalement, un autre inconvénient de cette méthode est lié à la résolution effective. Les raies de disparités sont éliminées et par conséquent le $SFDR$ augmente. Mais, par principe de la conservation d'énergie, le plancher du bruit du spectre de sortie du TIADC augmente et cause une dégradation du SNR et donc du nombre de bit effectif.

4.4.2 Méthode numérique adaptative

Cette méthode a été proposée dans [40], [41], [42] et [43]. Une approche similaire analogique a été publiée dans [44]. Dans cette section nous nous intéressons à la solution numérique en ligne [40]. Le principe de cette méthode vise à fixer les deux gains des deux convertisseurs à une même constante. Quant à la calibration de l'offset, le but est de faire tendre l'offset d'un CAN vers celui de référence. Nous détaillons dans la suite le principe de la correction des erreurs de gain et d'offset.

4.4.2.1 Principe de la calibration du gain

L'idée consiste à trouver une méthode de calibration qui fixe les différentes valeurs de gain à une seule constante pour tous les convertisseurs du TIADC. Si toutes les valeurs de gain des différents convertisseurs tendent vers la même constante, les raies spectrales de disparité des

gain disparaissent. Ainsi, calibrer un TIADC revient à calibrer un seul CAN et à appliquer aux différents convertisseurs la même méthode de calibration. La figure 4.20 montre le système adaptatif de calibration du gain d'un seul convertisseur.

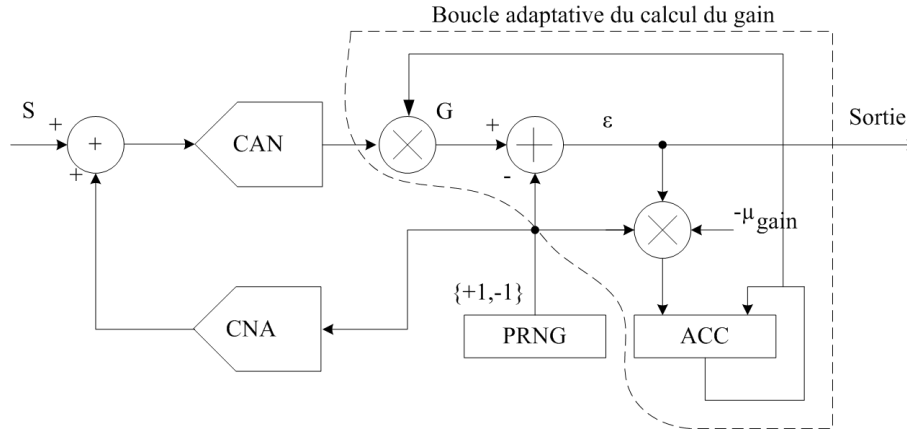


FIG. 4.20 – Calibration du gain pour un seul CAN.

Les blocs essentiels sont : un PRNG (Générateur de Nombres Pseudo Aléatoires), un CNA (Convertisseur Numérique Analogique) à 1 bit, le CAN sous calibration, un multiplieur numérique avec une variable de gain G déterminée dans la boucle adaptative et un accumulateur numérique (ACC).

La séquence N générée par le générateur de nombres pseudo aléatoires est binaire et peut être assimilée à un bruit blanc avec une moyenne nulle non corrélé avec le signal d'entrée S . Durant la calibration, la séquence N générée est convertie en un bruit analogique à l'aide du CNA 1 bit et ensuite additionnée au signal d'entrée. La même séquence est soustraite de la sortie, la différence ε est considérée comme la sortie finale du CAN calibré. Enfin, ε est multipliée par N , mise en échelle par un nombre négatif $-\mu_{gain}$ (pour que la valeur de ε ne tend pas vers l'infinie après accumulation) et accumulée pour déterminer la valeur de gain G dans la boucle de rétroaction.

Pour simplifier l'analyse, supposons que le CAN et le CNA soient représentés par leur modèles linéaires sous forme d'amplificateurs avec G_A et G_D comme valeurs de gain respectif. Considérons maintenant sur la figure 4.20 les deux chemins du générateur de nombres pseudo aléatoires à l'entrée du soustracteur qui calcule la valeur de ε . Si le gain du premier chemin qui est égale à $G_A \times G_D \times G$, n'est pas équivalent à celui du deuxième qui est égale à 1, le résultat de la soustraction présente un résidu de nombre aléatoire dans ε .

Dans ce système adaptatif, l'algorithme du *Least Mean Square* LMS est appliqué pour calculer

la valeur du gain G selon les équations 4.24 et 4.25 :

$$G(n+1) = G(n) - \mu_{gain} N(n) \varepsilon(n) \quad (4.24)$$

et

$$\varepsilon(n) = G_A G(n) S(n) + G_D G_A G(n) N(n) - N(n) \quad (4.25)$$

où n est l'indice du temps discret et μ_{gain} est le pas de l'algorithme. La soustraction de 4.25 et de 4.24 en utilisant $N^2(n) = 1$ donne

$$G(n+1) = G(n) + \mu_{gain} - \mu_{gain} G_D G_A G(n) - \mu_{gain} G_A G(n) N(n) S(n) \quad (4.26)$$

En régime établie le dernier terme de l'équation 4.26 tend vers zéro puisque le signal S est non corrélé avec la séquence N . Ainsi

$$\overline{G(n+1)} = \mu_{gain} + \overline{G(n)} (1 - \mu_{gain} G_A G_A) \quad (4.27)$$

Si la moyenne de la variable $G(n)$ converge, alors

$$\lim_{n \rightarrow \infty} \overline{G(n+1)} = \lim_{n \rightarrow \infty} \overline{G(n)} \quad (4.28)$$

Substituons 4.28 dans 4.27, cela donne

$$\overline{G(\infty)} G_A = \frac{1}{G_D} \quad (4.29)$$

Ainsi, la sortie du convertisseur calibré a une valeur de gain constante égale à $\frac{1}{G_D}$.

Appliquons cette méthode de calibration de gain dans les deux convertisseurs d'un TIADC conformément au processus de calibration de gain illustré dans la figure 4.21. Le système contient deux boucles identiques utilisant le même CNA pour forcer le gain de chaque CAN à $\frac{1}{G_D}$ en moyenne. Les deux convertisseurs présentent la même valeur de gain et par la suite les raies de disparité de gain doivent être éliminées.

4.4.2.2 Principe de la calibration d'offset

La figure 4.22 montre le diagramme adaptatif de la calibration d'offset dans le cas d'un TIADC à deux convertisseurs. Les séquences d'entrée du bloc de calibration sont les sorties des CAN après la correction du gain. Chacune de ces séquences contient la valeur du signal numérisé et une constante d'offset additive relative à chaque convertisseur. Soit V_{OS1} et V_{OS2} les offsets de chaque convertisseur respectivement. La variable d'offset O est ajoutée à la sortie du deuxième convertisseur, le résultat est soustrait à la sortie du premier convertisseur. La différence est mise

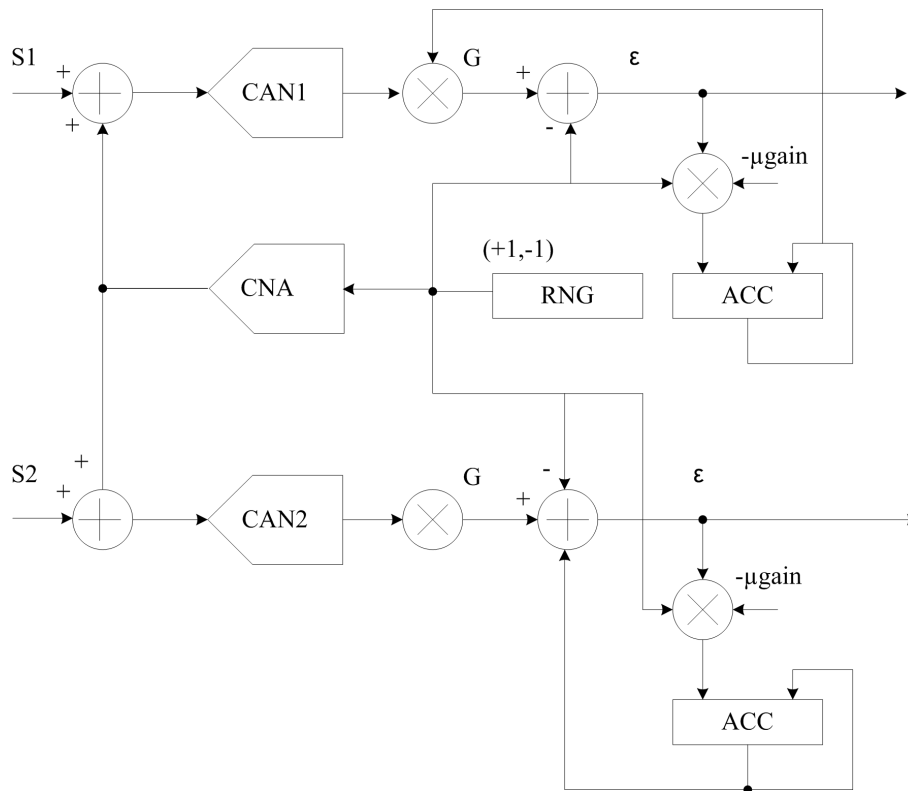


FIG. 4.21 – Calibration du gain pour un TIADC à deux CAN.

en échelle par la constante μ_{offset} pour déterminer O . Si le pas μ_{offset} est suffisamment petit, la moyenne de O converge vers

$$\bar{O} = V_{OS1} - V_{OS2} \quad (4.30)$$

Ainsi, après calibration les deux convertisseurs du TIADC présentent la même valeur d'offset, V_{OS1} .

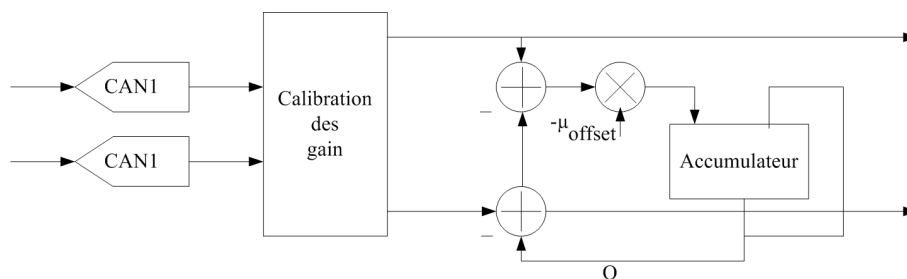


FIG. 4.22 – Calibration d'offset pour un TIADC à deux CAN.

4.4.2.3 Simulations

Le système de calibration de gain et d'offset est simulé dans le même environnement de description matérielle du modèle du TIADC. La description matérielle est relativement simple puisqu'il s'agit de blocs numériques simples : additionneurs, soustracteurs, registres et accumulateurs. Les opérations de divisions utilisées comportent une dividende sous forme d'un vecteur de bits signé et un diviseur sous forme de puissance de 2. Ceci revient à un décalage vers la droite de la dividende.

Pour les opérateurs de multiplication, plusieurs architectures de multiplieurs peuvent être utilisées comme les multiplieurs par additions successives, par additions et décalages, par décalages, par table de transcodage ou la multiplication directe ($A \times B$). Les trois premières architectures effectuent la multiplication de deux vecteurs de bits sur plusieurs périodes d'horloge. Par conséquent, pour une application d'entrelacement de convertisseur où la rapidité est l'avantage majeur, ces architectures ne seront pas employées. D'un autre côté, la multiplication séquentielle par table de transcodage n'est pas bien adaptée à cette méthode de correction. En effet, cette architecture ne peut pas contenir des variables génériques sur les tailles des vecteurs du multiplieur et du multiplicande. Donc, par élimination, le multiplieur utilisé dans cette méthode de correction est la multiplication directe.

Tous les opérateurs cités servent à la calibration d'offset dont nous donnons la description matérielle utilisant le langage VHDL-AMS en annexe C. Pour la correction du gain, ce sont les mêmes blocs qui sont utilisés avec le générateur de nombres pseudo aléatoires dont la description est détaillée dans l'annexe C. Les résultats de simulations sous formes de spectre de sortie sont listés dans le figure 4.23.

Les paramètres d'erreurs caractérisant les deux CAN sont identiques à ceux donnés dans les simulations de la méthode de calibration μ par FFT. Les constantes μ_{offset} et μ_{gain} sont fixées respectivement à 2^{27} et 2^{28} pour être conforme aux constantes données dans [40]. Les valeurs de ces constantes sont certes très importantes et constituent un inconvénient de la méthode, mais des valeurs inférieures ont été essayées en guise d'amélioration et le système devient instable. Pour des valeurs de μ_{offset} et μ_{gain} fixées à 2^{27} et 2^{28} , les amplitudes des composantes non-harmoniques de disparités d'offset et de gain ont subies une nette diminution, de 7 dB et de 13 dB respectivement, suite à la calibration.

4.4.2.4 Discussions

La calibration d'offset utilisée consiste en une accumulation de la différence des sorties de deux CAN. Le résultat de l'accumulation est additionné à l'une des deux voies du TIADC. En sortie, les deux convertisseurs ont le même offset. Ceci est justifié par les simulations : la composante

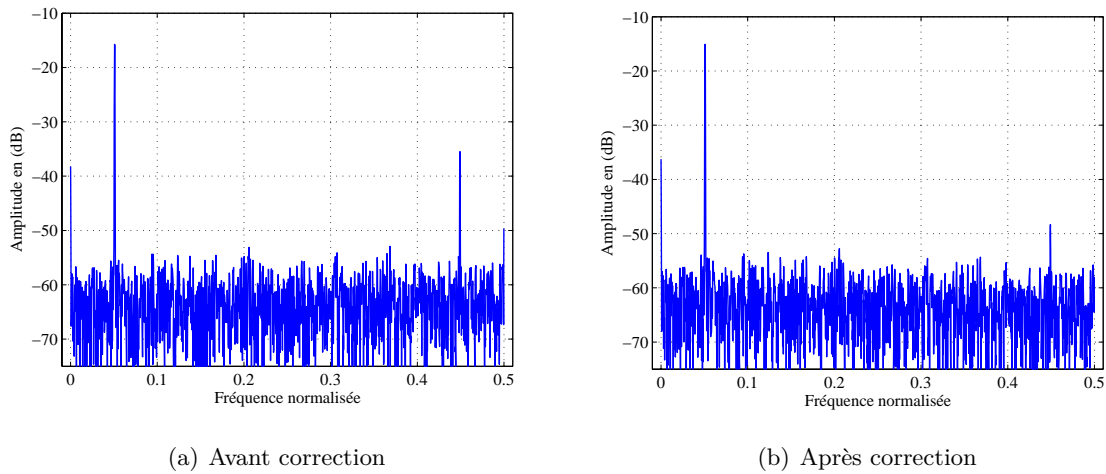


FIG. 4.23 – Spectres de sortie avant et après correction.

située à la fréquence de Nyquist est éliminée. Pour la correction du gain, la méthode proposée utilise, entre autres, un générateur de nombres aléatoires. Les sorties des deux CAN après correction du gain présentent la même valeur de gain. Par conséquent, la composante non-harmonique de la disparité de l'erreur du gain devrait être totalement éliminée. Dans les simulations présentées dans [40], l'amplitude de cette composante diminue de 10 dB pour un CAN de résolution 10 bits. Ceci vient du fait que la moyenne de la séquence de nombres aléatoires générée n'est pas nulle.

Il a été montré dans [40] que le nombre de multiplieurs dans la calibration pour un TIADC peut être réduit en remplaçant les deux multiplieurs multi bits par un seul à la sortie du générateur de nombres aléatoires. Ceci minimise la surface de silicium et la puissance consommée mais augmente le temps de convergence puisque la valeur de G_2 ne converge que si G_1 converge. Il a été montré aussi dans [45] que la méthode proposée assure la linéarité du système.

4.5 Conclusion

Dans ce chapitre nous avons exposé une liste non exhaustive de méthodes de calibration pour les convertisseurs analogique numérique. Nous avons cité dans une première partie les différentes classes de ces méthodes pour un seul convertisseur tout en insistant sur leurs avantages et leurs inconvénients. Il s'agit de méthodes analogiques et numériques d'un côté et de méthodes en ligne et en différé de l'autre. Dans la deuxième partie de ce chapitre nous nous sommes focalisés sur les méthodes de post traitement numérique pour la correction des erreurs de disparité dans les TIADC. Pour cela nous avons mis l'accent sur quatre méthodes de calibration dont deux sont en

différé et deux en ligne. Nous avons, à l'aide de simulations, comparé et évalué les performances de ces méthodes. Ainsi, ces méthodes ont été validées par des simulations sous Matlab et par des descriptions en langage VHDL-AMS. Nous présentons dans le tableaux 4.2 une comparaison se basant sur le principe de fonctionnement et sur l'évolution des paramètres spectraux.

Référence de la méthode	[34]	[37]	[39]	[40]
Type	en différé	en différé	en ligne	en ligne
Fréquence maximale	$(M - 1)f_s$	Mf_s	$\leq (M - 1)f_s$	Mf_s
Complexité approximative	++	+++	±	+
Évolution du SNR	10 dB	7 dB	-15 dB	2 dB
Évolution du SFDR	31 dB	24 dB	7 dB	13 dB
Temps de calibration	+	+++	±	+

TAB. 4.2 – Comparaison des méthodes de correction des TIADC.

Il est à noter que le traitement de l'erreur de retard d'ouverture ne parait que dans [37] et dans [39]. La première méthode est complexe, la seconde ne présente pas une nette amélioration des paramètres spectraux. Les méthodes proposées pour la correction présentent pour chacune autant d'avantages que d'inconvénients.

À travers l'étude des différentes méthodes de correction nous concluons qu'une méthode de correction qui se veut efficace doit garantir un facteur de mérite (FOM) élevé du TIADC. Pour se faire, la méthode de correction doit allier fréquence maximale de fonctionnement, précision, temps de calibration et complexité algorithmique et matérielle. Elle doit aussi traiter tous types de signaux, notamment les signaux non-stationnaires (ce n'est pas le cas dans les méthodes exposées dans [34], de [37] et de [40]).

Dans le chapitre suivant, nous essayons de répondre à ce besoin en proposant deux nouvelles solutions pour la correction des disparités des erreurs dans les TIADC.

Méthodes de correction proposées

Sommaire

5.1	Introduction	101
5.2	Méthode en différé	102
5.2.1	Principe	102
5.2.2	Validations	103
5.2.3	Discussions	105
5.3	Méthode en ligne	106
5.3.1	Principe de la méthode proposée	106
5.3.2	Le filtrage adaptatif	108
5.3.3	Les méthodes d'interpolation	113
5.3.4	Validations	118
5.3.5	Discussions	121
5.4	Amélioration de la solution proposée.	122
5.4.1	Le filtre NLMS	122
5.4.2	Le filtre NDLMS	125
5.5	Conclusion	128

5.1 Introduction

Après avoir présenté le principe d'un CAN, l'influence des erreurs de disparités sur les performances spectrales des TIADC et les méthodes de correction existantes dans les chapitres

précédents, nous nous intéressons dans ce chapitre à de nouvelles méthodes de correction. Cet intérêt est stimulé d'une part par l'insuffisance des méthodes existantes et d'autre part par la non-accessibilité à certaines méthodes. La méthode de correction à utiliser dépend d'abord de l'application visée. C'est la raison pour laquelle nous proposons deux types de méthodes de correction. La première méthode proposée est une méthode en différé, et se base sur des calculs de moyennes. Les erreurs traitées par cette méthode sont les erreurs d'offset et de gain. La deuxième méthode proposée est une méthode en ligne, et se base sur la notion de filtrage adaptatif et d'interpolation. Des améliorations concernant la rapidité et le traitement des signaux non stationnaires seront aussi analysées.

5.2 Méthode en différé

5.2.1 Principe

Il s'agit d'une méthode purement mathématique se basant sur le calcul des moyennes et l'extraction des paramètres d'erreurs [46]. Les erreurs en question sont les erreurs d'offset et de gain. Cette méthode est une méthode en différé et peut être utilisée pour corriger la disparité des erreurs d'un TIADC dans des applications de mesure. Pour cette méthode, la correction d'offset se fait séparément dans les différents canaux afin de fixer les valeurs d'offset à zéro. La correction du gain est réalisée en fixant une des voies comme référence et en faisant tendre la valeur des gains des autres CAN vers celle du gain de la référence. Le processus de correction, illustré dans la figure 5.1, se fait conformément aux étapes suivantes :

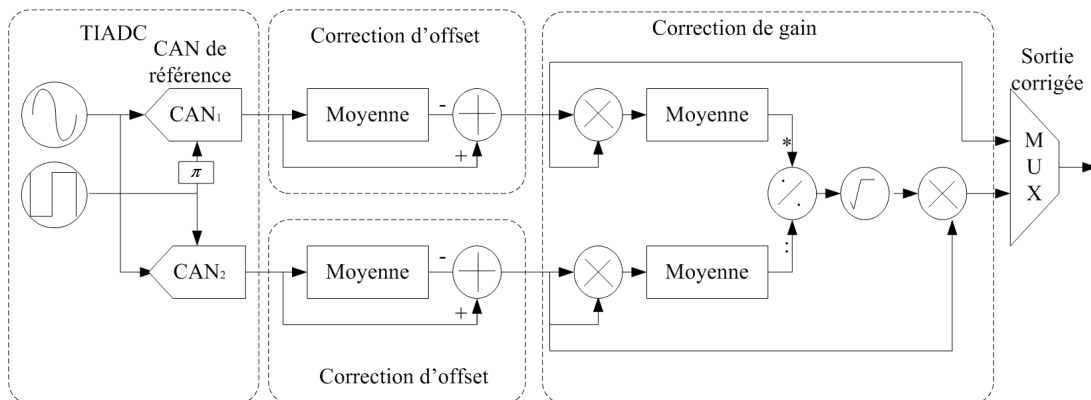


FIG. 5.1 – Principe de la correction pour deux CAN.

- charger les données de sorties des deux CAN ;
- calculer les moyennes pour extraire la valeur des offsets ;
- par soustraction, les sorties ne présentent pas d'erreur d'offset ;

- extraire le rapport de gain $\frac{g_1}{g_i}$, ($2 \leq i \leq M$), par moyennage et calcul mathématique ;
- par multiplication, les sorties de tous les CAN présentent le même gain, identique à celui du CAN de référence.

L'idée de correction d'offset existe dans la littérature depuis longtemps. Il s'agit de la même méthode présentée dans [40]et [41]. Pour un signal de sortie d'un CAN de forme

$$s(n) = O + G \times A \cos(2\pi f_0 n T_s + \varphi) + eq(n) \quad (5.1)$$

et en supposant que le bruit de quantification est négligeable devant l'amplitude du signal convertie, la moyenne de $s(n)$ renvoie une valeur estimée \hat{O} de l'offset. La soustraction $s(n) - \hat{O}$ compense l'erreur d'offset dans le convertisseur en question. L'application du même traitement pour les différentes voies du TIADC séparément corrige les erreurs d'offset et par conséquent élimine l'effet de la disparité d'offset.

La nouveauté que nous apportons par cette méthode est l'utilisation des moyennes pour la correction du gain. Dans le cas de deux CAN entrelacés, le signal obtenue après correction d'offset, $s_1(n) = s(n) - \hat{O}$, sert à son tour pour la correction du gain. La moyenne de $s_1(n)$ est égale à zéro. Par contre, la multiplication de $s_1(n)$ par lui même donne

$$s_2(n) = (A.G)^2 + (A.G)^2 \times \cos(2\pi f_0 n T_s + \varphi) \quad (5.2)$$

La moyenne du signal $s_2(n)$ donne

$$f(G) = (A.G)^2 \quad (5.3)$$

L'application de la même procédure sur les deux CAN séparément donne $f(G_1) = (A.G_1)^2$ et $f(G_2) = (A.G_2)^2$. Par division de ces deux constantes, nous obtenons le rapport élevé au carré des deux gains.

Les sorties des deux CAN après correction d'offset présentent deux valeurs de gain différentes, la multiplication d'une des deux voies par le rapport des deux gains fixe le gain du deuxième canal à celui du premier.

5.2.2 Validations

Valider une méthode de correction d'offset et de gain c'est d'abord vérifier si les composantes non-harmoniques engendrées par les disparités de ces erreurs sont éliminées ou réduites. Les spectres des figures 5.2, 5.3 et 5.4 montrent l'état de ces raies avant la correction, après la correction d'offset et après la correction du gain.

Les simulations ont été réalisées en utilisant le modèle du TIADC développé en VHDL-AMS et exposé en annexe B. Nous avons utilisé une amplitude du signal d'entrée égale à 1 V pour une pleine échelle de 4 V. Les CAN utilisés pour l'entrelacement sont de résolution 8 bits. Les

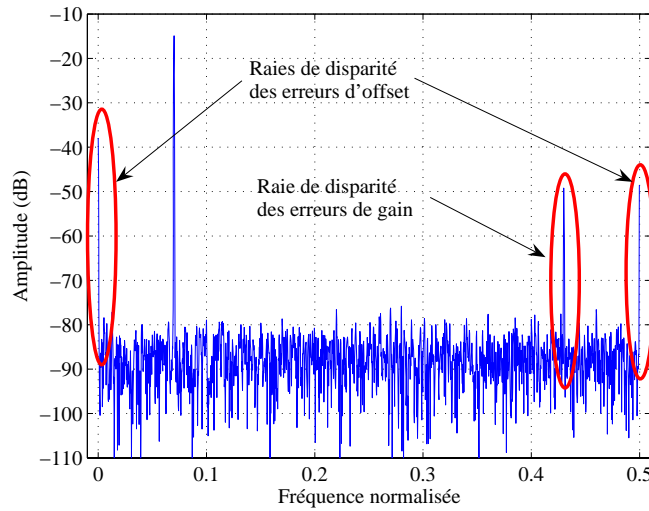


FIG. 5.2 – Spectres de sortie avant la correction

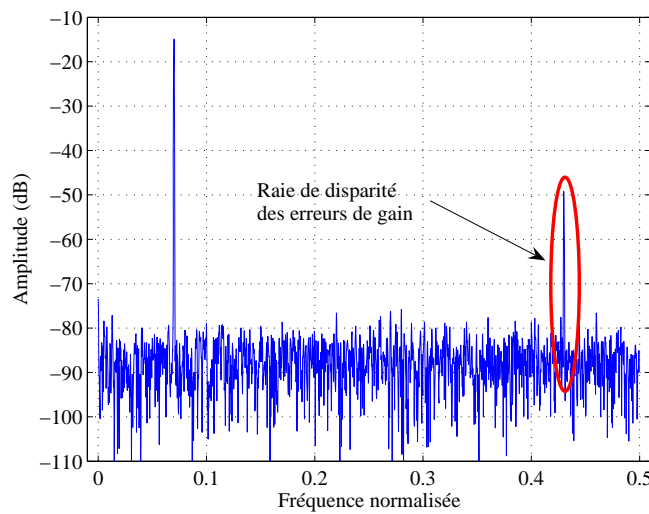


FIG. 5.3 – Spectres de sortie après la correction d'offset

fréquences du signal d'horloge et du signal d'entrée respectent le théorème de la cohérence et sont respectivement autour de 500 MHz et de 70 MHz . Les valeurs d'erreurs d'offset sont $O_1 = 2.10^{-2}$ V et $O_2 = 5.10^{-2}$ V. Les valeurs des erreurs de gain sont $G_1 = 1 + 10^{-2}$ et $G_2 = 1 + 5.10^{-2}$. Le jitter inséré par les deux CAN est de 0.09 de la période d'échantillonnage. Le nombre de point est fixé à 4096.

Après correction d'offset, les valeurs de \hat{O}_1 et de \hat{O}_2 sont respectivement de 0.023142 V et 0.053698 V. Les valeurs de gain sont $\hat{G}_1 = 1 + 98.10^{-4}$ et $\hat{G}_2 = 1 + 48.10^{-3}$. Ces valeurs estimées

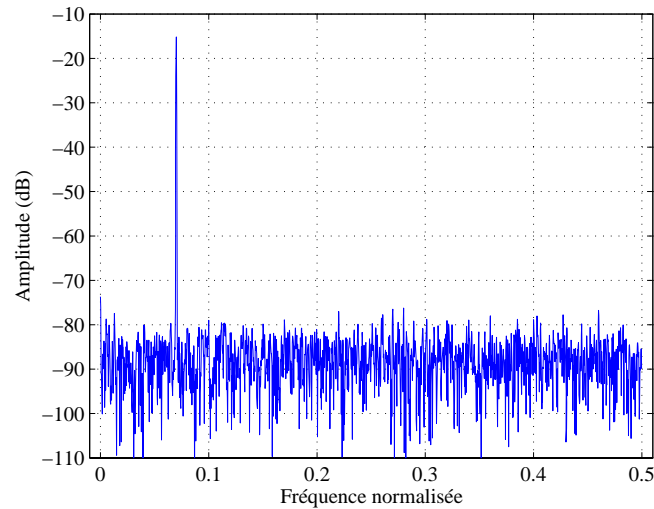


FIG. 5.4 – Spectres de sortie après la correction de gain

sont très proches des paramètres d'erreurs insérés. En soustrayant les valeurs d'offset calculées de la sortie des CAN et en divisant les valeurs de gain calculées des sorties des CAN après correction d'offset, les raies de disparités sont complètement éliminées.

Par conséquent, la valeur du *SFDR* augmente et le graphique de la figure 5.5 montre l'évolution du *SFDR* en fonction de l'amplitude du signal d'entrée. Pour des faibles amplitudes du signal d'entrée par rapport à la pleine échelle, l'amélioration du *SFDR* est autour de 35 dB.

5.2.3 Discussions

La méthode proposée se base sur un calcul mathématique utilisant des règles classiques de la trigonométrie. Le processus de correction se fait après avoir enregistré les données de sorties des CAN, c'est donc une méthode de calibration en différé. Nous avons remarqué dans la figure 5.4 que les raies de disparités des erreurs d'offset et de gain sont éliminées. La calibration en question n'utilise aucune information sur les signaux d'entrée et d'horloge. Il s'agit d'une calibration aveugle. Un autre avantage consiste en la possibilité de l'application de cette méthode pour n'importe quel nombre de CAN. En effet, la calibration des offsets pour chaque convertisseur se fait indépendamment de l'autre et celle du gain se fait par rapport à un convertisseur de référence.

La réalisation matérielle de la méthode proposée n'est pas compliquée. En effet, les blocs utilisés pour cette méthode sont principalement des opérateurs d'addition et de multiplication et leur réalisation matérielle est simple. Mais la méthode en question est une méthode en différé qui sert principalement pour des applications de mesures et par conséquent la correction peut se

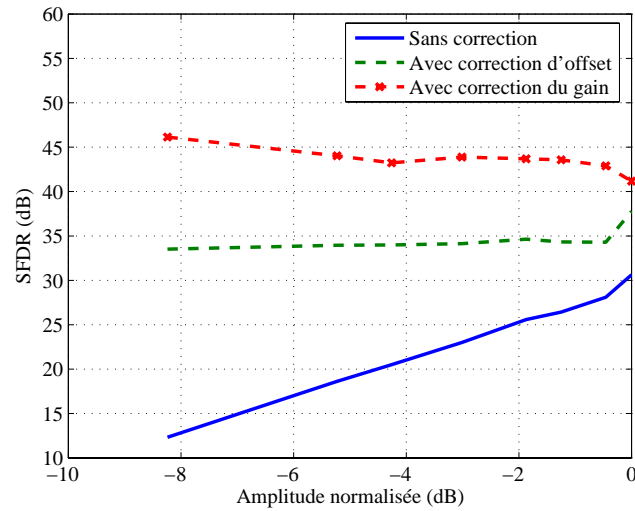


FIG. 5.5 – Évolution du SFDR

faire en logicielle.

Un inconvénient majeur de cette méthode vient du fait que la correction ne se fait que pour les erreurs statiques. Une insertion d'un jitter ou d'un retard d'ouverture ne perturbe pas la correction des disparités de gain et d'offset, comme le cas de la méthode de calibration par FFT exposée au chapitre précédent, mais l'effet de ces erreurs temporelles reste sans correction. Une méthode de correction efficace devrait tenir compte de ces erreurs afin de les corriger.

Néanmoins, cette méthode peut servir de référence pour valider la correction d'offset et de gain pour d'autres solutions notamment pour des méthodes en ligne.

5.3 Méthode en ligne

Dans cette partie, il s'agit de proposer une méthode de correction en ligne qui pourra être utilisée dans des applications à temps réel. La méthode ne doit pas prendre en considération les caractéristiques du signal d'entrée. En d'autres termes, la méthode doit s'adapter à toutes formes du signal d'entrée.

5.3.1 Principe de la méthode proposée

Le but de la méthode est de réduire les disparités entre les deux canaux du TIADC. Puisque l'amplitude et la fréquence des harmoniques dépendent des caractéristiques du signal d'entrée et du signal d'horloge, un filtre à coefficients fixes dans le temps ne suffit pas. Pour cela nous utilisons un filtrage adaptatif qui a pour rôle d'approcher les caractéristiques des signaux des

CAN. Une sortie d'un CAN est prise comme entrée à filtrer, la deuxième est utilisée comme référence. La voie filtrée tend vers le signal de référence et la différence entre les deux signaux est suffisamment faible pour réduire les raies de disparités.

Cependant, un problème de principe se pose : les deux signaux de sortie du TIADC ne sont pas sur un même front d'horloge. Cette différence de front d'horloge est vu par le filtre adaptatif comme erreur de décalage. Pour minimiser l'erreur résiduelle, le filtre adaptatif élimine ce décalage et il en résulte que les deux signaux sont sur un même front d'horloge, le front du signal de référence. Ce fait est contradictoire par rapport au principe de l'entrelacement temporelle.

Le signal de référence doit être disponible sur les deux fronts du signal d'horloge pour effectuer le filtrage adaptatif d'un côté et le multiplexage temporel de l'autre côté. Deux possibilités sont alors envisageables :

1. échantillonner le signal de référence à une fréquence double de sa fréquence d'échantillonnage maximale, ce qui est absurde ;
2. les valeurs des échantillons sont disponibles sur un front d'horloge, il faut les estimer sur le deuxième front d'horloge, cela s'appelle une interpolation.

En effet, comme le montre la figure 5.6, l'entrée de référence au filtre adaptatif n'est plus la sortie d'un des deux CAN, mais désormais la sortie interpolée d'un des deux CAN. Les signaux d'entrée, de référence et de sortie du filtre sont sur le même front d'horloge. Le multiplexeur du TIADC a donc pour entrées la sortie du filtre adaptatif et la sortie du CAN utilisée comme référence du filtre, mais sans interpolation. Le schéma bloc de la figure 5.6, illustre le principe de la méthode de correction. Les blocs $D1$ et $D2$ sont des registres à décalage utilisés pour compenser le retard consommé par l'opération d'interpolation ($D1$) et par le filtre adaptatif ($D2$). Nous proposons dans les sections suivantes de rappeler la notion du filtrage adaptatif et

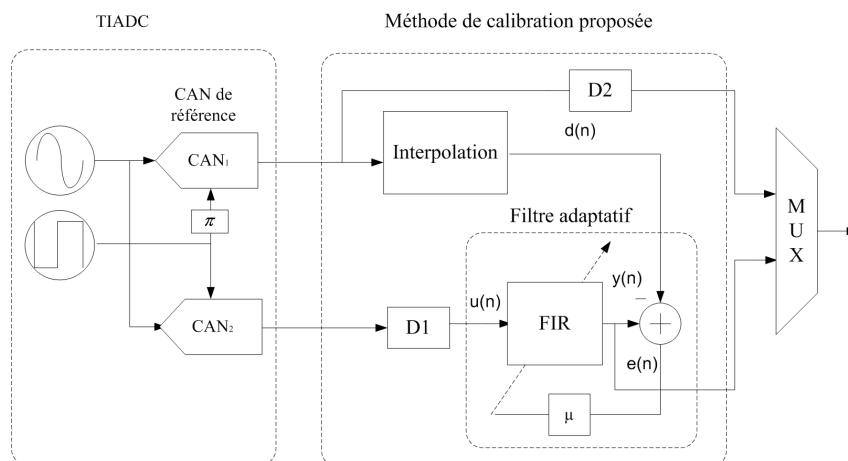


FIG. 5.6 – Principe de la méthode de correction.

la notion de l'interpolation afin de mieux cerner le principe de la méthode.

5.3.2 Le filtrage adaptatif

5.3.2.1 Principe et applications

Depuis les années 60, les méthodes adaptatives en traitement de signal ont connu un essor considérable, grâce au développement du traitement numérique du signal et à l'augmentation constante de la puissance des calculateurs, permettant ainsi l'implémentation en temps réel d'algorithmes de plus en plus sophistiqués à des cadences de plus en plus rapides [47].

Les filtres adaptatifs sont des filtres linéaires dont les coefficients sont modifiés au cours du temps. Ces modifications sont assurées par un algorithme adaptatif qui minimise l'énergie du signal d'erreur e suivant un critère donné.

Ces filtres sont utilisés dans plusieurs applications dont l'annulation d'écho, l'égalisation et l'identification. Le filtrage optimale de Wiener, considéré comme la base de tous les filtres adaptatifs, s'installe dans le contexte de ces applications. La figure 5.7 détaille le principe de fonctionnement de ce type de filtre. Les signaux utilisés sont : $u(n)$ processus aléatoire observé, $x(n)$ processus

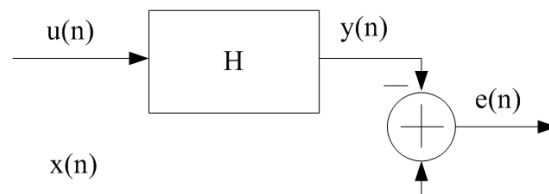


FIG. 5.7 – Filtrage optimal de Wiener.

désiré, $y(n)$ estimation de $x(n)$ et $e(n)$ erreur d'estimation.

La mise en oeuvre d'un filtre (estimateur) optimal de Wiener demande la connaissance des caractéristiques du signal et du bruit. Cela implique également que ces caractéristiques soient stables au cours du temps, ce qui n'est pas le cas en pratique. Le filtrage adaptatif a pour objet d'approcher ces filtres optimaux. Pour cela, les coefficients de la réponse impulsionnelle du filtre sont adaptés en fonction de l'erreur par une boucle de retour comme le montre la figure 5.8.

Cette adaptation nécessite une séquence d'apprentissage et une stratégie de mise à jour des coefficients du filtre dont l'objectif est la minimisation d'une erreur. La stratégie de mise à jour des coefficients définit le type du filtre adaptatif. De ce fait, il existe deux grandes familles de filtres adaptatifs :

- les filtres LMS (*Least Mean Square*), la mise à jour des coefficients nécessite la connaissance des derniers échantillons du signal à filtrer et de l'erreur ;

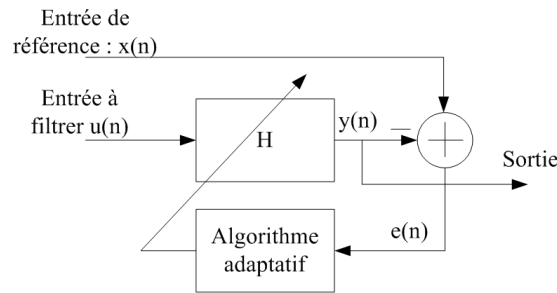


FIG. 5.8 – Structure des filtres adaptatifs.

– les filtres RLS (*Recursive Least Squares*), le calcul de l'adaptation des coefficients utilise, contrairement au filtre LMS, tout le passé de l'information ;

Pour le filtre LMS, l'adaptation des coefficients se fait instantanément en multipliant le pas d'adaptation, le signal à filtrer et l'erreur. Pour le filtre RLS, la correction consiste en un produit de deux facteurs : l'estimation de l'erreur ($e(n)$) et le vecteur de gain $k(n)$ qui correspond au produit du signal à filtrer et de l'inverse de la matrice d'autocorrélation du signal à filtrer. L'inverse de cette matrice d'autocorrélation constitue à la fois l'avantage (précision accrue) et l'inconvénient (temps de calcul important) de ce filtre. Pour pouvoir comparer les deux algorithmes, nous visons une application qui consiste à annuler les erreurs d'offset et de gain d'un signal sinusoïdal à la sortie d'un CAN. Le signal de référence est une sinusoïde pure, alors que le signal à l'entrée du filtre présente des erreurs d'offset et de gain.

Une première simulation consiste à déterminer le temps nécessaire pour l'adaptation des coefficients en se basant sur l'évolution de l'erreur résiduelle. Pour le même ordre du filtre, le temps requis pour le filtre RLS est nettement inférieure que celui requis pour le filtre LMS, figure 5.9. Une deuxième simulation consiste à calculer, non pas le temps nécessaire pour la phase d'adaptation (simulations de la figure 5.9), mais le temps d'exécution de l'opération du filtrage adaptatif pour différents ordres du filtre (L).

Les résultats de simulation indiquent que le temps requis pour l'exécution de l'opération du filtrage dans le cas du filtre RLS est en moyenne deux fois plus importants que dans le cas de l'utilisation du filtre LMS : bien qu'il existe des méthodes avancées pour l'optimisation du calcul de l'inverse de la matrice d'autocorrélation (LEMMA [48]), le filtre RLS consomme plus de ressources et de temps de calcul que le filtre LMS. Le tableau 5.1 confirme ce fait en donnant le nombre de multiplieurs et d'additionneurs utilisés par chaque filtre. Encore une fois, le choix de l'algorithme adaptatif dépend des contraintes de l'application. Dans le cadre de nos études, le but est d'utiliser un algorithme adaptatif pour minimiser les disparités entre les différents canaux d'un TIADC. La solution d'utiliser un filtre RLS réduit le taux d'échantillonnage du

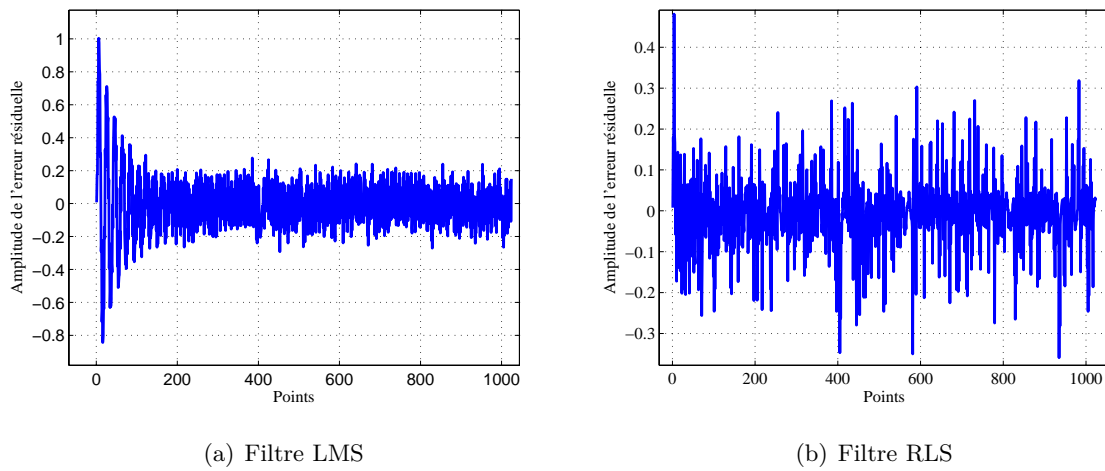


FIG. 5.9 – Evolution de l'erreur résiduelle pour les filtres LMS et RLS.

Filtre	Multiplieur	Additionneur
LMS	$2L$	$2L$
RLS	$3L^2 + 4L$	$2L^2 + 2L$

TAB. 5.1 – Comparaison de la complexité des deux filtres LMS et RLS.

système, par conséquent elle est mise à l'écart. En plus, un choix judicieux du pas d'adaptation du filtre LMS permet de réduire le temps de convergence sans aucun coût supplémentaire.

5.3.2.2 Le filtre LMS

Ce filtre est le plus populaire et le plus utilisé dans les applications industrielles en raison de son extrême simplicité de mise en oeuvre, de sa robustesse aux erreurs de calcul et de la diversité des contextes de son utilisation. La structure du filtre LMS est exposée dans la figure 5.10. Dans le processus du filtrage adaptatif, le filtre utilisé est un filtre à réponse impulsionnelle finie (FIR). Bien que les filtres à réponse impulsionnelle infini (IIR) présentent le même gabarit pour des ordres inférieurs, ils ont l'inconvénient d'avoir une phase non-linéaire. Plusieurs travaux ont été menés pour la linéarisation de la phase de ces filtres utilisant des techniques différentes sans pour autant garantir une stabilité ou empêcher des problèmes de saturations, [49], [50] et [51]. C'est la raison pour laquelle nous avons choisi un filtre FIR.

Le but de l'algorithme de filtrage est d'effectuer des opérations sur le signal $u(n)$ (entrée du filtre) pour fournir une sortie dont la différence avec le signal $x(n)$ (signal de référence) soit minimisée. Cette minimisation est obtenue en calculant les coefficients du filtre pour chaque

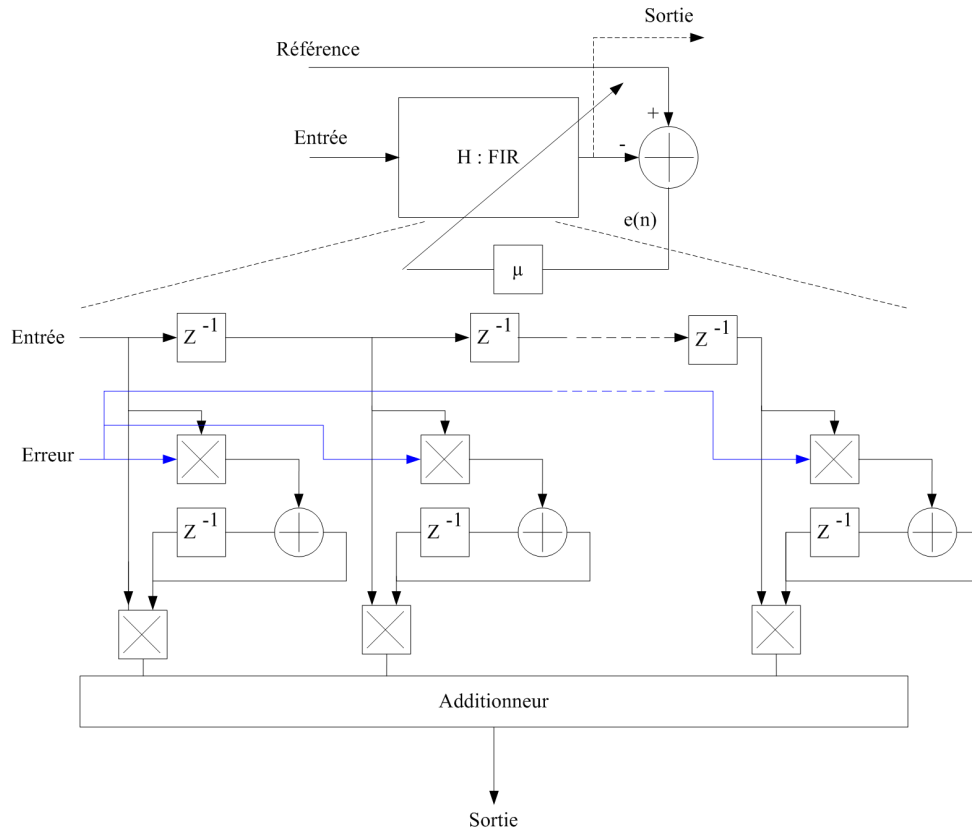


FIG. 5.10 – Structure du filtre LMS.

nouvel ensemble de données, de référence et de signal reçu. Ainsi, en supposant qu'à l'indice n , n ensembles de données aient été reçus, les coefficients du filtre FIR adaptatif sont représentés par le vecteur $H(n)$ qui minimise la fonction de coût $J(n)$ définie par :

$$J(n) = \sum_{k=1}^n (x(k) - H^t(n)U(k))^2 \quad (5.4)$$

où $U(k)$ est le vecteur colonne d'éléments $(u(k), u(k-1), \dots, u(k+1-L))$. Il a été démontré dans [48], [52] et dans [2] que l'erreur en sortie du système est de la forme :

$$e(n+1) = x(n+1) - H^t(n)U(n+1) \quad (5.5)$$

Cette erreur est appelée erreur à priori puisque les indices $n+1$ de l'erreur sont calculés avec les indices n des coefficients. La mise à jour des coefficients est faite par l'équation :

$$H(n+1) = H(n) + \mu U(n+1)e(n+1) \quad (5.6)$$

L'algorithme ainsi obtenu est appelé algorithme du gradient car la quantité $-U(n+1)e(n+1)$ représente le gradient de la fonction $\frac{1}{2}e^2(n+1)$, c'est à dire la valeur instantanée de l'erreur

quadratique. Ainsi, l'algorithme converge en moyenne vers une solution optimale. Le critère de minimisation correspondant est le critère des moindres carrés moyens. Dans le cas où l'erreur résiduelle est nulle après convergence, le pas d'adaptation est borné par la relation :

$$0 < \mu < \frac{2}{L\sigma_u^2} \quad (5.7)$$

où σ_u^2 désigne la puissance du signal d'entrée et L l'ordre du filtre FIR. Il est à signaler aussi que le pas et le temps d'adaptation du filtre adaptatif sont inversement proportionnels. Dans plusieurs applications on cherche souvent la plus grande rapidité d'adaptation et l'on souhaite donc donner au pas d'adaptation la plus grande valeur possible. La figure 5.11 montre l'évolution

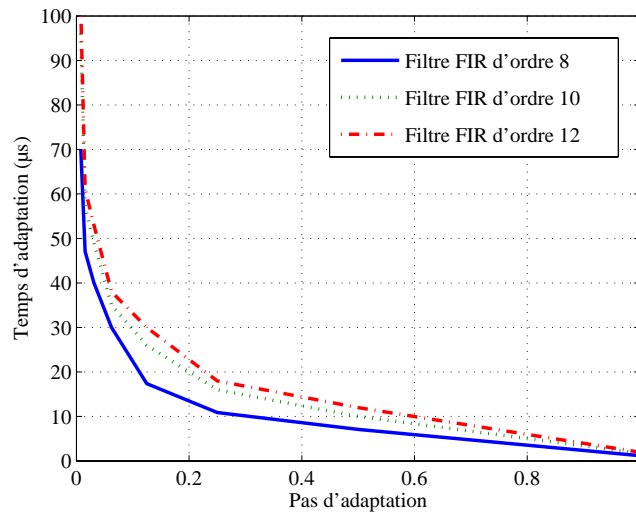


FIG. 5.11 – Évolution du temps d'adaptation.

du temps d'adaptation par rapport au pas d'adaptation pour différents ordres du filtre FIR et pour des signaux d'entrée et de référence sinusoïdaux et d'amplitudes unitaires. Nous remarquons que pour des pas d'adaptation supérieurs à 0.25 le temps d'adaptation est nettement plus faible (de l'ordre de $10 \cdot 10^{-6}$ s sur une station Sun Ultra 5). Ainsi, l'ordre du filtre FIR pour des signaux issus d'un CAN de résolution res est limité par l'équation 5.8.

$$\frac{4\sqrt{2}}{A} < L < \frac{2^{res-\frac{1}{2}}}{A} \quad (5.8)$$

Un ordre 8 est alors suffisant pour la convergence des coefficients du filtre pour un CAN de résolution supérieure à 3 bits. Il est à noter que les limitations de la taille des coefficients et des données internes contribuent à l'augmentation de l'erreur résiduelle totale.

5.3.3 Les méthodes d'interpolation

5.3.3.1 Définition

Pour évaluer une fonction, il est fréquent de ne disposer que de ses valeurs sur un ensemble fini de points. Dans le cas le plus simple d'une fonction à une variable, ces points sont souvent disposés sur un réseau régulier unidimensionnel. Il est souvent nécessaire de pouvoir évaluer cette fonction en dehors de cet ensemble de points. Si le point à évaluer se situe à l'intérieur d'un intervalle élémentaire constitué de deux points consécutifs du réseau, la procédure d'approximation de la fonction par une fonction plus simple s'appelle l'interpolation. Si le point à évaluer se situe en dehors des bornes du réseau, la procédure d'approximation s'appelle une extrapolation. La mise en place d'une interpolation pour une fonction évaluée initialement sur un réseau consiste à optimiser la fonction approchante en utilisant l'ensemble des données [53]. Une fonction d'interpolation doit posséder les propriétés suivantes :

1. elle doit se confondre avec la fonction à interpoler sur son ensemble de définition ;
2. elle doit être la plus régulière possible ;
3. elle doit être décrite par une quantité finie d'informations.

Dans la suite de cette section nous détaillons quelques méthodes d'interpolation pour extraire leurs avantages et leurs inconvénients.

5.3.3.2 Interpolation au plus proche voisin

L'interpolation au plus proche voisin (PPV) est certainement la méthode d'interpolation la plus simple. Cette méthode consiste à attribuer à chaque point cible la valeur de son voisin le plus proche. Si les voisins sont équidistants, on doit faire un choix arbitraire (à gauche ou à droite) du point interpolé. Cette méthode est peu satisfaisante dans le sens où elle introduit souvent de larges paliers lors de l'interpolation comme le montre la figure 5.12. Néanmoins, elle a le mérite de conserver la dynamique du signal et de ne pas introduire des états intermédiaires.

5.3.3.3 Interpolation linéaire

Dans le cas d'une interpolation linéaire (bilinéaire en deux dimensions), on considère que la courbe est localement une droite dont on détermine l'équation à l'aide des coordonnées des deux points p_1 et p_2 de coordonnées respectifs (x_1, y_1) et (x_2, y_2) les plus proches du point recherché. L'interpolation linéaire consiste à affecter au point interpolé la moyenne pondérée des valeurs prises par les deux points les plus proches. D'un point de vue mathématique, l'interpolation linéaire consiste à trouver la fonction $f(x)$ définie sur l'intervalle $[x_1, x_2]$ par :

$$f(x) = y_1 + (y_2 - y_1)x \quad (5.9)$$

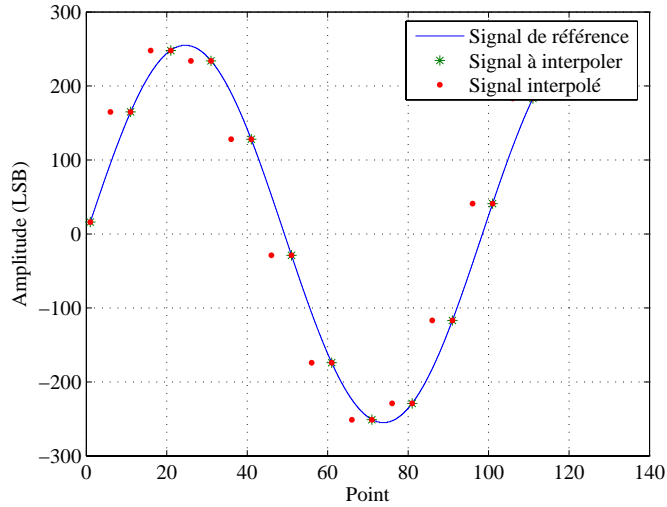


FIG. 5.12 – Interpolation au plus proche voisin.

Le but de l’interpolation consiste à évaluer le signal sur les fronts descendants du signal d’horloge. Ainsi, pour des points à interpoler correspondant aux instants nT_s , les points interpolés sont définies aux instants $(n + \frac{1}{2}) T_s$.

D’un point de vue structurel, nous modélisons l’interpolation linéaire de deux façons, conformément à la figure 5.13. Le modèle de la figure 5.13 (a), utilise un registre pour ne pas avoir de

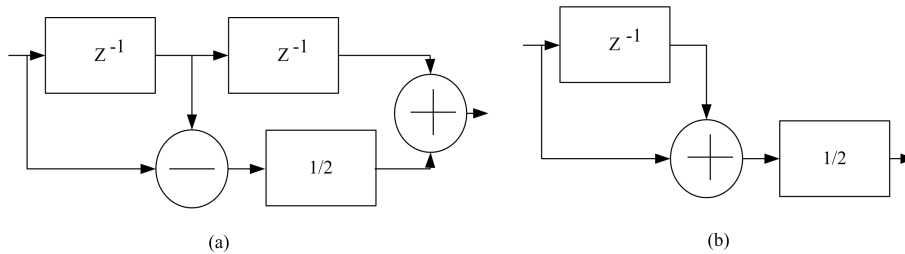


FIG. 5.13 – Modèles d’une interpolation linéaire.

décalage sur l’opérateur d’addition. Ce registre est prévu pour compenser le temps consommé lors de la division binaire. Le modèle de la figure 5.13 (b) est une représentation plus simplifiée de l’interpolation linéaire. Les résultats de simulation de ce modèle sont reportés dans la figure 5.14. En résumé, il s’agit d’une méthode simple et rapide à la fois. Mais l’erreur d’interpolation a des conséquences importantes dans certains cas. Le signal de sortie peut présenté des discontinuités. Pour remédier à ce problème, l’interpolation en cosinus est une méthode qui utilise un facteur de lissage sous forme d’un cosinus. L’équation 5.9 devient :

$$f(x) = y1 + (y2 - y1) \times \frac{(1 - \cos(\pi \times x))}{2} \tag{5.10}$$

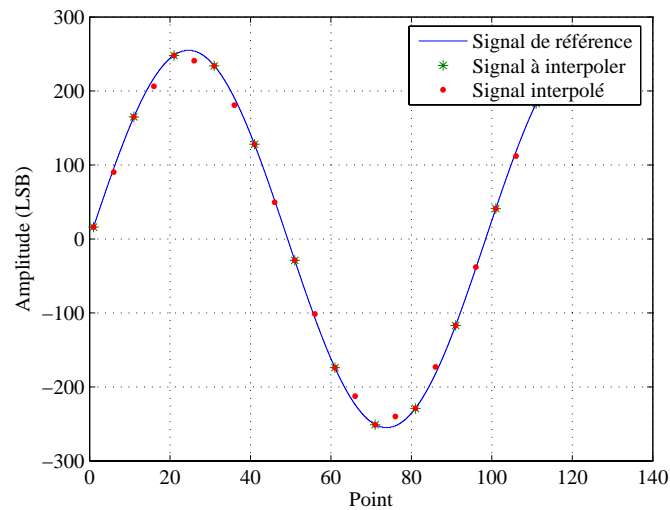


FIG. 5.14 – Interpolation linéaire.

Grâce au facteur de lissage, les discontinuités aux niveaux des points recherchés d'interpolation peuvent être réduites.

5.3.3.4 Interpolation cubique

Elle consiste à adjoindre bout à bout (entre deux points d'interpolation successifs) des polynômes du troisième degré avec des conditions de continuité de second ordre entre deux courbes polynomiales. Ceci permet d'obtenir des courbes d'interpolation régulières. Comme son nom l'indique, on utilise ici une équation cubique pour modéliser localement la courbe. Quatre points sont nécessaires pour évaluer la fonction qui remplace la courbe discrète. Ainsi, pour interpoler entre y_1 et y_2 , il faut utiliser les deux points de chaque côté du segment, soit y_0 , y_1 , y_2 et y_3 . En fonction des conditions de continuité utilisées, la forme de la fonction cubique peut varier et donner une interpolation différente :

- L'interpolation de type spline cubique, par exemple, est une forme d'interpolation cubique où le polynôme d'interpolation est défini entre les points du segment à interpoler sans faire appel aux points proches du segment [54].
- L'interpolation polynomiale (par Spline convolutive ou par B-Spline), consiste à trouver le polynôme unique P , de degré n ($n \geq 4$) qui passe par les $n + 1$ points à interpoler. Toutefois, pour déterminer les coefficients de ce polynôme, il est nécessaire d'inverser une matrice d'ordre $(n + 1) \times (n + 1)$: un calcul lourd de l'ordre de $\frac{2}{3}n^3$ opérations.
- D'autres formes d'interpolation cubique nettement plus efficaces utilisent une base de polynômes lagrangienne ou newtonienne pour obtenir une matrice respectivement diagonale ou

triangulaire.

En terme de précision, toutes les formes d'interpolation cubiques sont comparables [55]. L'inconvénient majeur de ces méthodes avancées consiste en leur temps d'interpolation élevé [56]. La fonction d'interpolation s'écrit donc sous la forme :

$$f(x) = a_0x^3 + a_1x^2 + a_2x + a_3 \quad (5.11)$$

avec

$$\begin{aligned} a_0 &= y_1 + y_3 - (y_0 + y_2) \\ a_1 &= 2(y_0 - y_1) + y_2 - y_3 \\ a_2 &= y_2 - y_0 \\ a_3 &= y_1 \end{aligned} \quad (5.12)$$

Ainsi, l'interpolation cubique d'ordre 0 équivaut à une interpolation par le plus proche voisin. L'interpolation cubique d'ordre 1 revient à une interpolation linéaire. Pour l'ordre 2, l'interpolation cubique (utilisant 3 points de calcul) est moins utilisée pour des raisons de dissymétrie. D'un point de vue structurelle, l'interpolation cubique est une description hiérarchisée composée d'éléments numériques de base : registre à décalage, additionneurs, soustracteurs et un multiplieur par 2 qui consiste à un décalage à gauche. Les résultats de simulation de l'interpolation cubique d'ordre 4 sont reportés dans la figure 5.15.

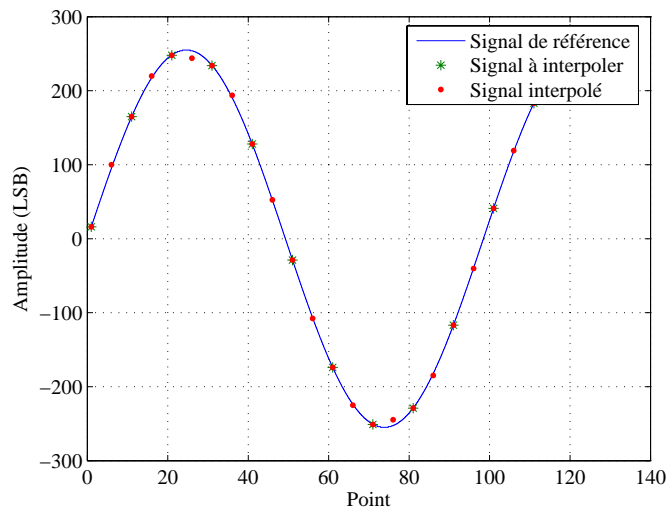


FIG. 5.15 – Interpolation cubique.

5.3.3.5 Choix d'une méthode d'interpolation

Dans le cadre de nos travaux sur la correction des erreurs des TIADC, la méthode d'interpolation doit être simple en terme de nombre d'opérations et précise pour garantir une certaine fidélité avec le signal d'origine. Pour pouvoir comparer ces différentes méthodes, un modèle a été établi pour calculer l'erreur engendrée par l'interpolation, (figure 5.16).

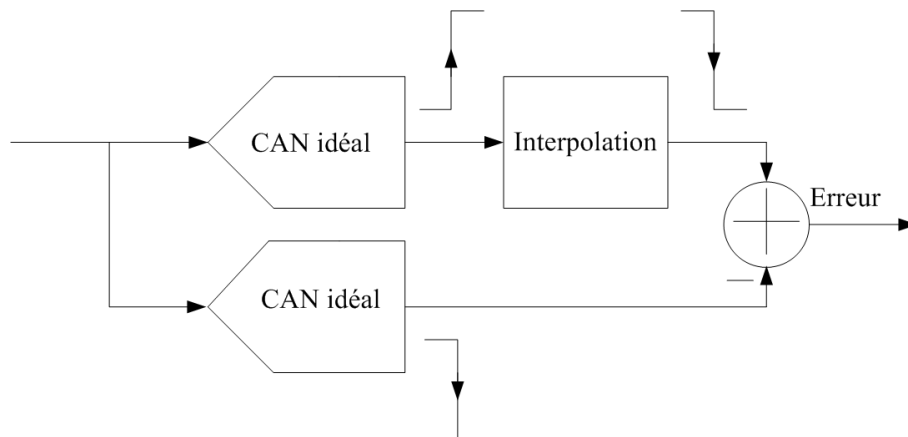


FIG. 5.16 – Modèle de calcul de l'erreur d'interpolation.

La sortie du CAN (supposé idéal pour l'instant) évolue sur les fronts montants de l'horloge d'échantillonnage. Cette sortie subit une interpolation pour estimer ses valeurs sur les fronts descendants de la même horloge. Ainsi, l'erreur d'interpolation n'est plus que la soustraction de cette sortie et de celle d'un CAN idéal évoluant sur les fronts descendants de l'horloge.

La figure 5.17 illustre l'amplitude de cette erreur pour un signal d'entrée de forme sinusoïdale utilisant différentes méthodes d'interpolation. Le nombre de points utilisés est égale à 1024. Pour estimer la valeur de l'erreur à chaque point, nous calculons la moyenne, pour chaque point, sur 100 acquisitions. Pour des fréquences d'entrée faibles par rapport à la fréquence d'échantillonnage, toutes les méthodes se valent : pour un CAN de résolution 8 bits et un rapport de $\frac{f_s}{f_0}$ égale 50 les pertes en SNR sont de 3.10^{-2} dB pour l'interpolation linéaire et de $1.8.10^{-2}$ dB pour l'interpolation spline. Cet écart augmente proportionnellement avec la résolution du convertisseur. Pour des fréquences d'entrées proches de celles de Nyquist, l'écart en terme de perte en SNR augmente en faveur de l'interpolation spline.

Une analyse de temps d'exécution de l'opération d'interpolation montre que les méthodes cubiques nécessitent 3 à 4 fois plus de temps que les méthodes linéaires ou PPV, [56]. Cette différence de rapidité est causée par le nombre d'opérations employées pour l'interpolation. Ainsi, dans le cas de la correction d'erreurs pour les TIADC, les méthodes d'interpolation cubique ou celles d'ordres supérieurs ne sont pas adaptées puisqu'elles causent une réduction de la cadence

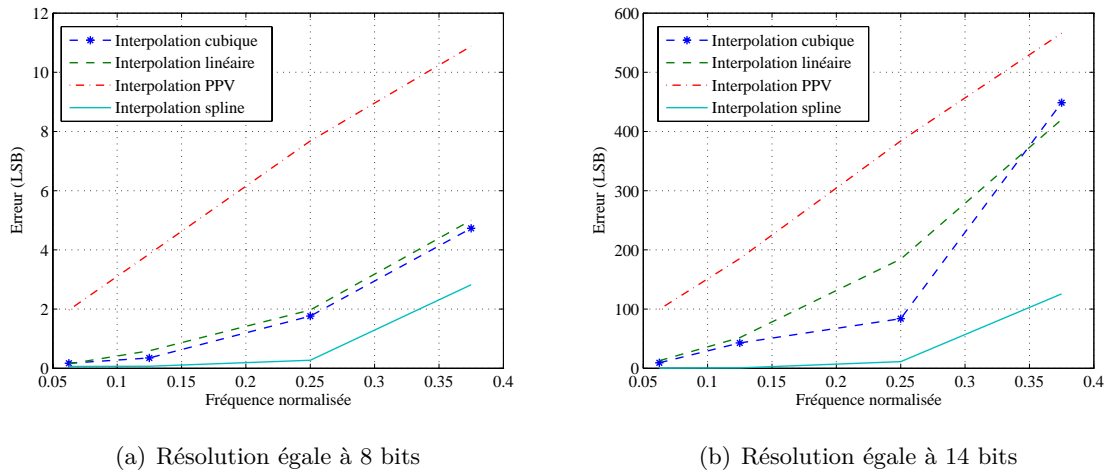


FIG. 5.17 – Évaluation de l'erreur d'interpolation.

d'échantillonnage du système. L'interpolation que nous avons choisi d'utiliser dans le reste de nos travaux est l'interpolation en cosinus.

5.3.4 Validations

La méthode proposée est une méthode adaptative en ligne. Elle se base sur le principe du filtrage adaptatif utilisant le filtre LMS. Une interpolation est nécessaire pour être en cohérence avec le principe de l'entrelacement. La méthode d'interpolation utilisée est de type interpolation en cosinus, le filtre adaptatif est un filtre LMS composé d'un filtre FIR d'ordre 8.

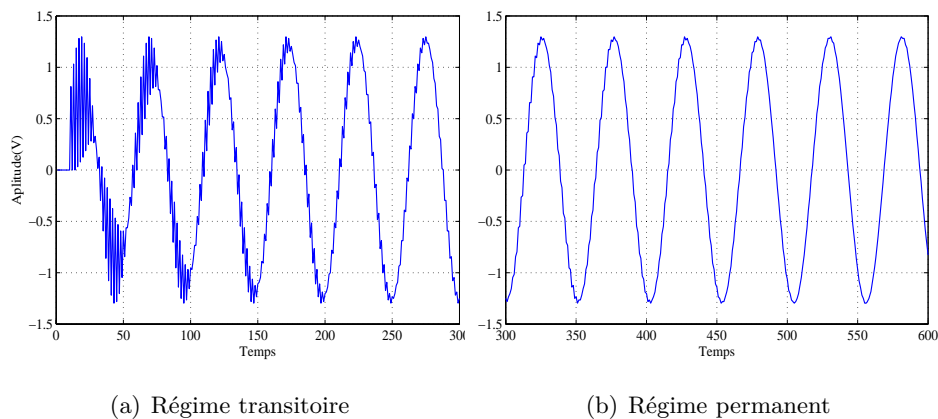


FIG. 5.18 – Sortie temporelle du TIADC.

La sortie temporelle du TIADC, représentée dans figure 5.18, montre la coexistence de deux régimes. Une première phase d'adaptation (figure 5.18.a) qui correspond au régime transitoire du système. La sortie du convertisseur à corriger part de zéro, et s'adapte ensuite par rapport au signal de référence. Après adaptation, (régime permanent, figure 5.18.b) les sorties des deux convertisseurs présentent des caractéristiques très proches. L'évolution de la sortie du CAN à corriger est montrée dans la figure 5.19.

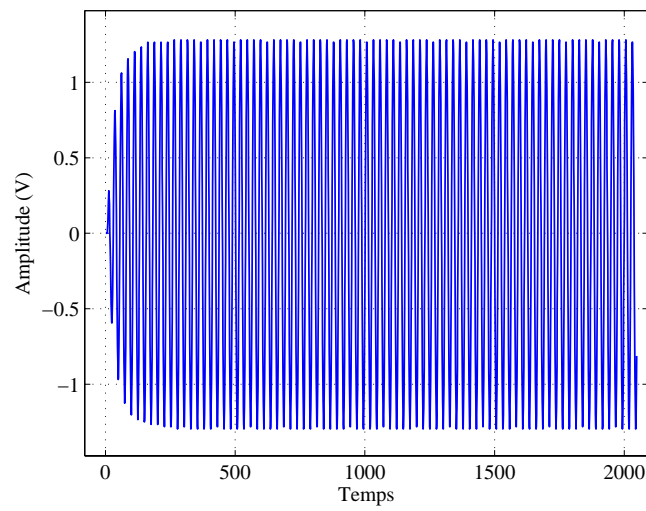


FIG. 5.19 – Évolution de la sortie du filtre FIR.

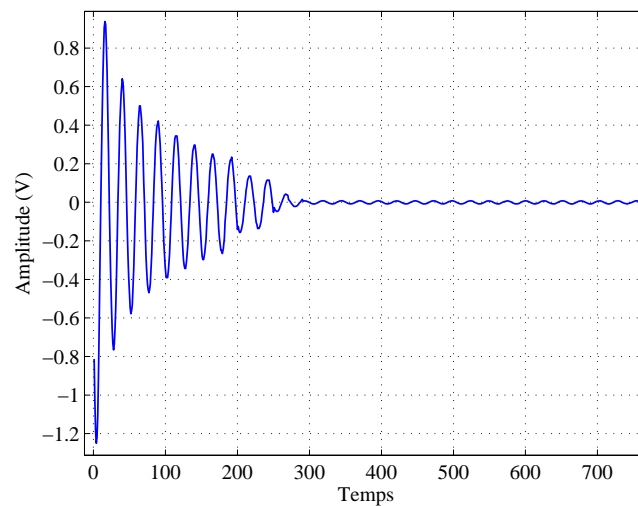


FIG. 5.20 – Évolution de l'erreur résiduelle.

Durant la phase d'adaptation, l'erreur résiduelle diminue au cours du temps jusqu'à sa mini-

misation, (figure 5.20). Le temps d'adaptation est géré par le pas de calcul du filtre LMS, μ . L'évolution du temps d'adaptation en fonction du pas de calcul est montrée dans la figure 5.11. Naturellement, le temps d'adaptation augmente avec l'ordre du filtre FIR et diminue lors de l'augmentation du pas de calcul. Au niveau spectral, les figures 5.21 et 5.22 montrent que l'amplitude de la raie de disparité est réduite de 16 dB pour une disparité de gain égale à 10% de l'amplitude du signal d'entrée et pour une différence de retard d'ouverture égale à 10% de la période d'échantillonnage.

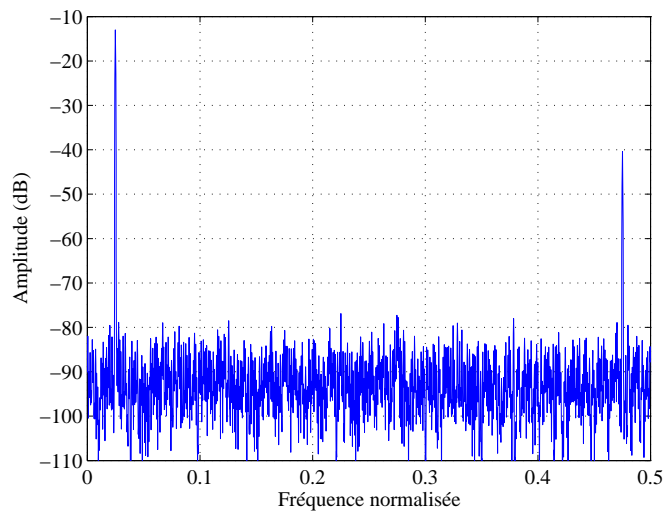


FIG. 5.21 – Spectre de sortie avant la correction.

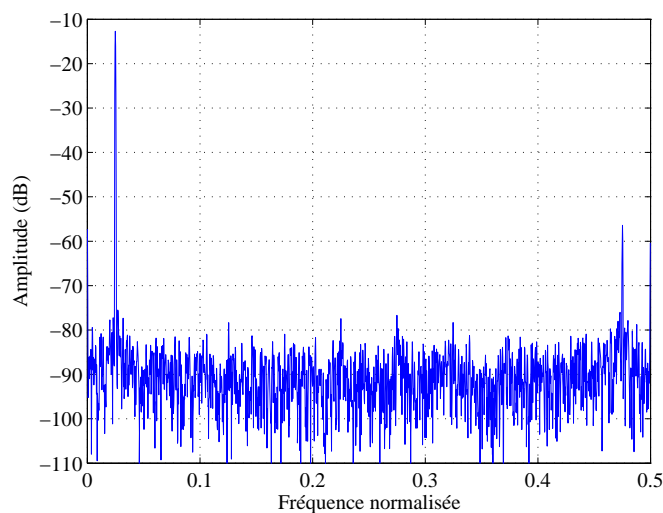


FIG. 5.22 – Spectre de sortie après la correction.

5.3.5 Discussions

La validation de la méthode a été réalisé pour des signaux stationnaires dans la section précédente. Mais, lors d'un changement brusque et important en fréquence ou en amplitude du signal à filtrer les coefficients du filtre changent sans aboutir à une convergence. Les raies de disparités des erreurs statiques et temporelles ne sont pas réduites et le *SFDR* se dégrade. Ce paramètre sert énormément pour l'évaluation de la qualité spectrale des signaux dans les applications de télécommunications, qui nécessite une méthode de correction en temps réel. Les signaux mis en jeux dans ces applications sont généralement des signaux sinusoïdaux qui varient en fréquence et en amplitude au cours du temps. La méthode de correction devrait, par conséquent, être validée pour ce type de signaux non stationnaires.

Cependant, le filtre LMS n'est pas adapté pour les signaux non stationnaires dont la fréquence et l'amplitude varient rapidement [47], [2]. La figure 5.23 montre le résultat de conversion avec et sans correction pour un signal d'entrée d'amplitude et de fréquence variables. Les agrandissements sur la figure montrent la non convergence du filtre LMS lors du changement d'amplitude ou de fréquence. Même pour certains types de signaux stationnaires comme le cas d'un signal en

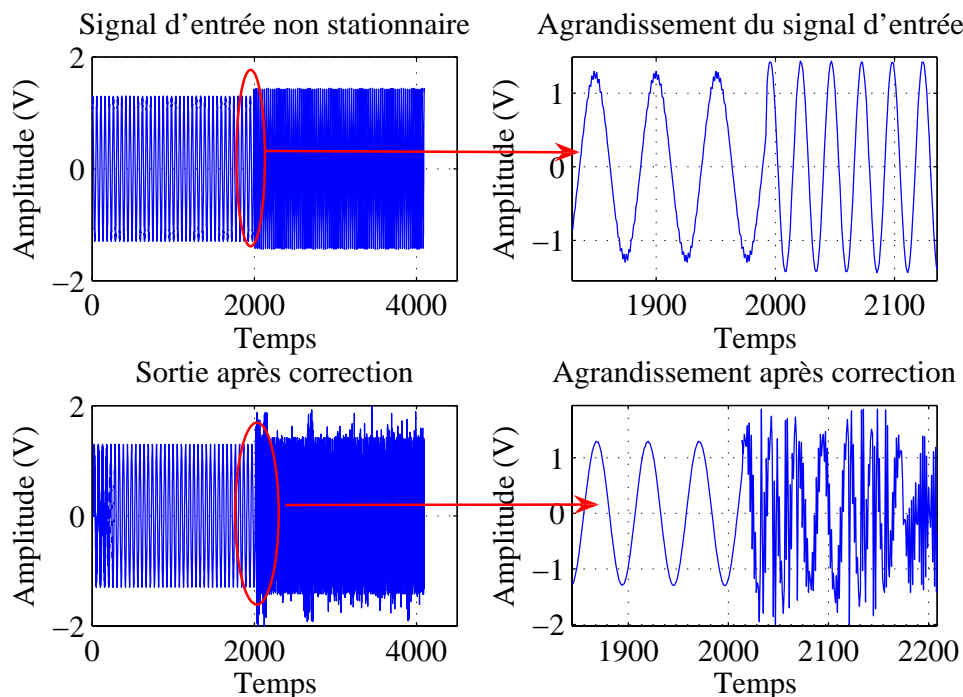


FIG. 5.23 – Sortie du TIADC avec et sans correction pour des signaux non stationnaires.

mode double ton (figure 5.24), [57], la méthode proposée ne corrige pas les erreurs de disparités. En effet, les paramètres d'adaptation des coefficients du filtre dépendent de l'énergie du signal

à adapter. La non convergence s'explique par l'indépendance entre le pas d'adaptation constant du filtre LMS et les caractéristiques du signal à filtrer.

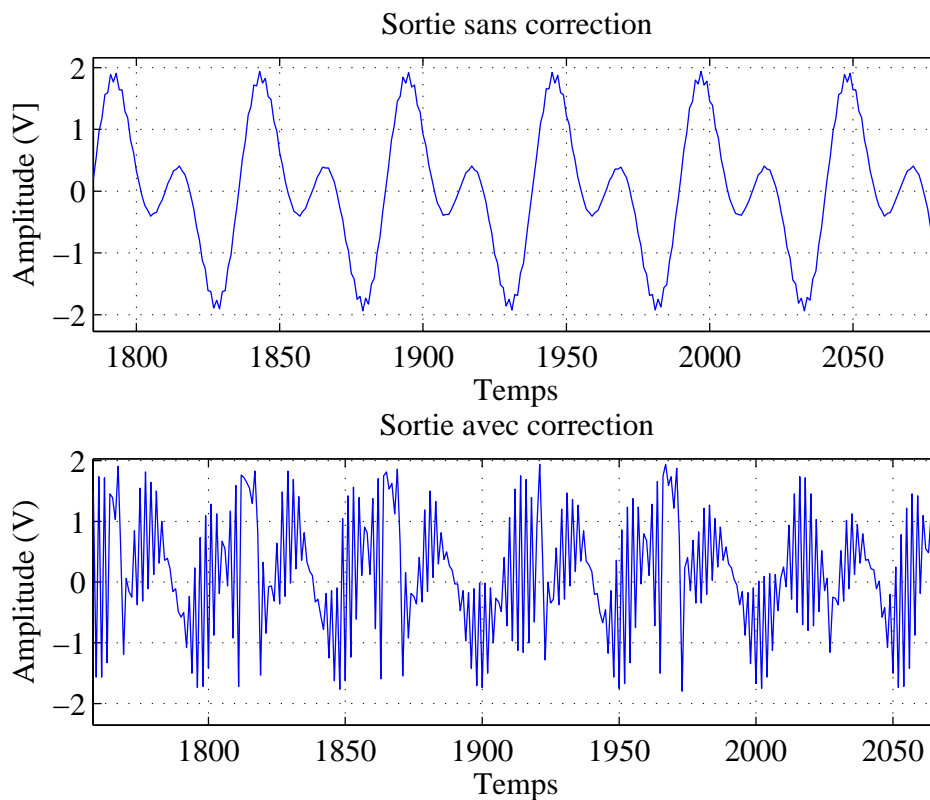


FIG. 5.24 – Sortie du TIADC avec et sans correction pour un signal dual ton.

5.4 Amélioration de la solution proposée.

5.4.1 Le filtre NLMS

5.4.1.1 Principe

L'amélioration présentée dans cette section vise à faire fonctionner la méthode proposée pour des signaux non stationnaires. Dans ce cas, le filtre adaptatif doit suivre l'évolution des paramètres de ces signaux. Cependant, un retard s'introduit lors de l'estimation des paramètres et entraîne une erreur de poursuite. C'est pourquoi, le supplément de l'erreur quadratique moyenne a deux composantes : l'erreur du gradient et l'erreur de poursuite.

La non stationnarité peut affecter le signal d'entrée au filtre et le signal de référence, ou l'un de ces deux signaux. Dans notre cas, la non stationnarité affecte les deux signaux à la fois. Le pas d'adaptation doit tenir compte de la variation de l'énergie des signaux au cours du temps.

Le filtre résultant est appelé filtre NLMS (Normalized LMS). Le filtre NLMS est géré par les mêmes équations que le filtre LMS. Seule l'équation 5.6 concernant l'adaptation des coefficients du filtre change de forme pour tenir compte de la dépendance du pas d'adaptation avec l'énergie du signal à adapter :

$$H(n+1) = H(n) + \overbrace{\frac{\mu}{X^t(n+1)X(n+1)}}^{\mu_{NLMS}} X(n+1)e(n+1) \quad (5.13)$$

Le produit scalaire $X^t(n+1)X(n+1)$ est une forme d'estimation de la puissance du signal. Ainsi, une variation de la puissance du signal X est suivi d'un changement du pas d'adaptation des coefficients du filtre.

5.4.1.2 Validations

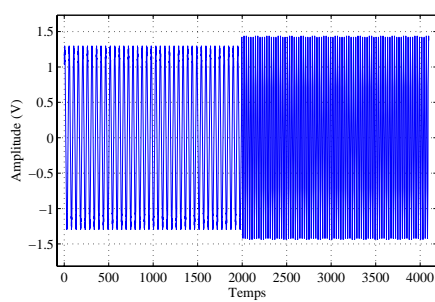
Pour des signaux stationnaires, le filtre NLMS a le même rôle que le filtre LMS. Les résultats de simulation sont les mêmes que celles présentées dans les figures 5.22 et 5.21. Pour des signaux non stationnaires, les résultats de simulation sont reportés dans les figure 5.25. Dans le cas de signaux non stationnaires, nous présentons dans la figure 5.25.a le signal de sortie d'un TIADC sans correction. Ce signal change de fréquence (de f_{in} à $2 \times f_{in}$) et d'amplitude (de A à $1.1A$) au cours du temps. La figure 5.25.b, montre le signal de sortie corrigé. Ici nous remarquons la présence d'une zone d'apprentissage entourée sur le graphique. Pour montrer l'effet de la correction, une analyse spectrale est établie. Les figures 5.25.c et 5.25.d montrent les deux spectres des deux phases du signal de sortie sans correction (spectres des signaux séparés de la figure 5.25.a) . Après correction, les figures 5.25.e et 5.25.f, montrent la réduction de l'amplitude des raies de disparité.

En mode double ton l'application du filtre NLMS permet la réduction des raies de disparité, figure 5.26. En effet, dans ce cas de figure, le signal à convertir est présenté sous forme de deux sinusoïdes à des deux fréquences différentes, f_{in1} et f_{in2} . Les raies de disparités de gain et de retard temporel sont localisées aux fréquences $f_s - \frac{f_{in1}}{2}$ et $f_s - \frac{f_{in2}}{2}$. Les amplitudes de ces deux raies sont réduites de 10 dB et de 16 dB respectivement.

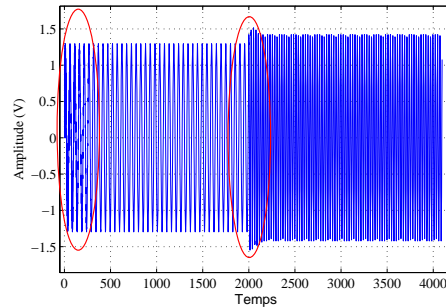
5.4.1.3 Discussions

Le problème de convergence du filtre LMS est en rapport avec la nature non stationnaire des signaux. La solution étant de lier le pas d'adaptation du filtre à la puissance du signal à filtrer. Le filtre est nommé filtre NLMS. Le fonctionnement est validé par simulations dans le cas de signaux non stationnaires et en mode double ton.

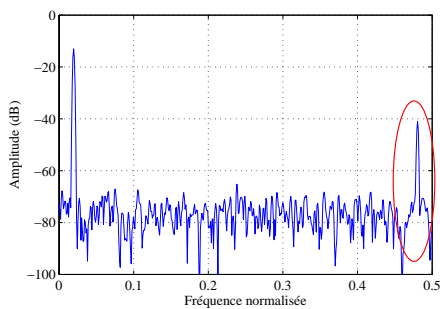
Une deuxième amélioration peut être ajoutée au système pour avoir plus de rapidité. En effet



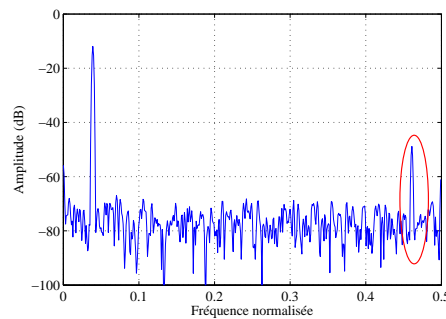
(a) Signal temporel avant correction.



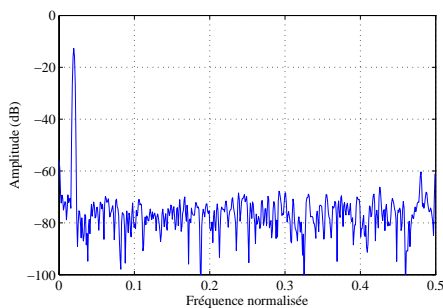
(b) Signal temporel après correction.



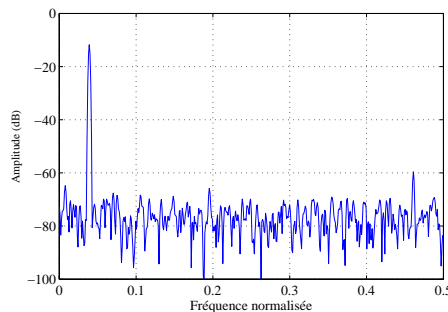
(c) Spectre de la phase 1 avant correction.



(d) Spectre de la phase 2 avant correction.



(e) Spectre de la phase 1 après correction.



(f) Spectre de la phase 2 après correction.

FIG. 5.25 – Sortie du filtre NLMS pour un signal non stationnaire.

nous constatons une différence nette lors de la simulation du TIADC avec ou sans correction : le temps de simulation augmente considérablement en présence de correction. Ceci sera vérifié par estimation de la fréquence maximale de fonctionnement, dans le prochain chapitre. Dans le cas des TIADC, il est évident que la solution proposée ne devrait, en aucun cas, réduire le débit du système. Une amélioration de la rapidité de la structure est exposée dans la section suivante.

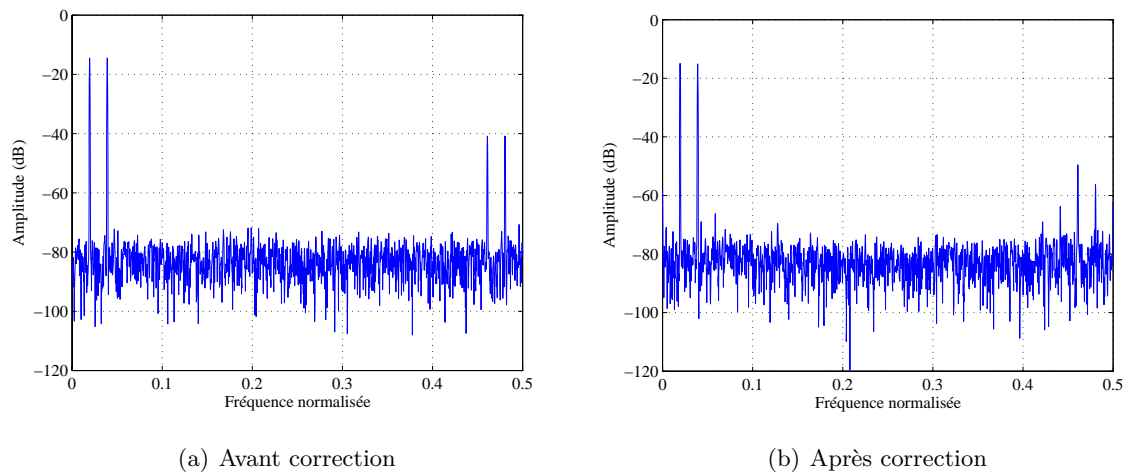


FIG. 5.26 – Sortie du filtre NLMS pour un signal en mode dual ton.

5.4.2 Le filtre NDLMS

5.4.2.1 Principe

Cette section a pour but de contourner le problème de rapidité du filtre NLMS. La structure du filtre LMS est détaillée dans la figure 5.10. Une représentation par étage du filtre FIR utilisée dans le filtre LMS est montrée dans la figure 5.27. Bien que la représentation soit en étage et

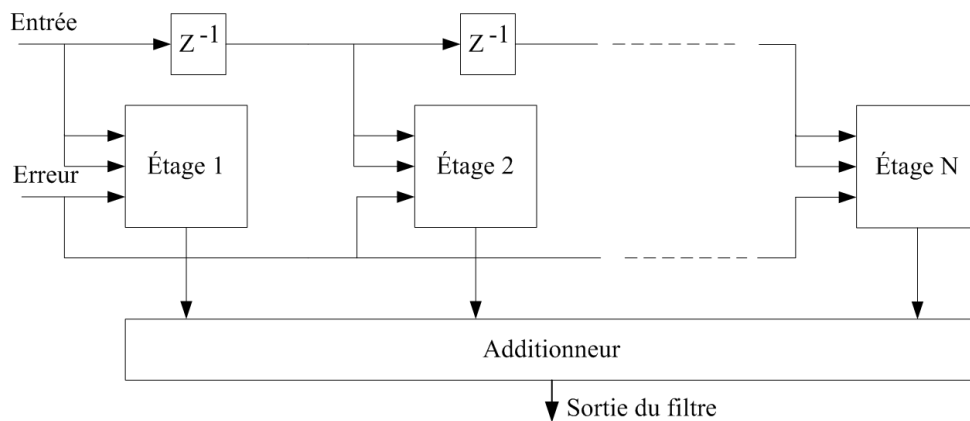


FIG. 5.27 – Structure du filtre FIR.

que la structure du filtre soit parallèle, le calcul ne se fait pas de façon parallèle. En effet, tous les étages du filtre utilisent, au même temps, la valeur n de l'erreur pour le calcul de la valeur $n + 1$ de la sortie. Tous les calculs se font à la même étape et par conséquent, le temps de cycle devient très long.

Le filtre DLMS (Delayed LMS) [58], [59] et [60], est une solution efficace pour remédier à ce fait. La théorie et l'étude de stabilité et de la convergence de ce filtre sont détaillées dans [61], [62], [63] et [64]. Des utilisations de ce filtre dans des applications audio [65] et dans l'égalisation de canaux [66] montrent la robustesse de ce type de filtre.

Le principe du filtre DLMS consiste à introduire des délais sur l'erreur avant l'entrée de chaque étage. Les étages du filtre FIR deviennent totalement indépendants et le temps de cycle est par conséquent réduit. Puisque nous ajoutons un élément de retard sur l'erreur avant l'entrée de chaque étage, nous devons ajouter le même élément de retard sur le signal d'entrée du filtre. Les blocs représentés par des Z^{-1} de la figure 5.27 cèdent leur place pour des blocs en représentés par des Z^{-2} . La structure complète du filtre DLMS est présentée dans la figure 5.28. Le bloc D consiste à introduire D éléments de retard sur l'entrée à filtrer. Généralement, le nombre de retard, D, est égale à l'ordre du filtre FIR. Les opérations d'addition sur les sorties des étages du filtre FIR se font sous forme d'arbre d'addition. Des éléments de retard sont donc nécessaires.

Un agrandissement sur la structure d'un étage est montré dans la figure 5.28. Ces étages sont appelés des *Processor Module* (PM). Les sorties de chaque PM sont utilisées comme entrées du PM suivant. Mais les éléments de retard utilisés font en sorte que les calculs sont totalement indépendants. Dans un cycle d'horloge, tous les PM font le même type de calcul, mais sur des entrées différentes. La structure du filtre DLMS est donc parallèle. Ce parallélisme implique une

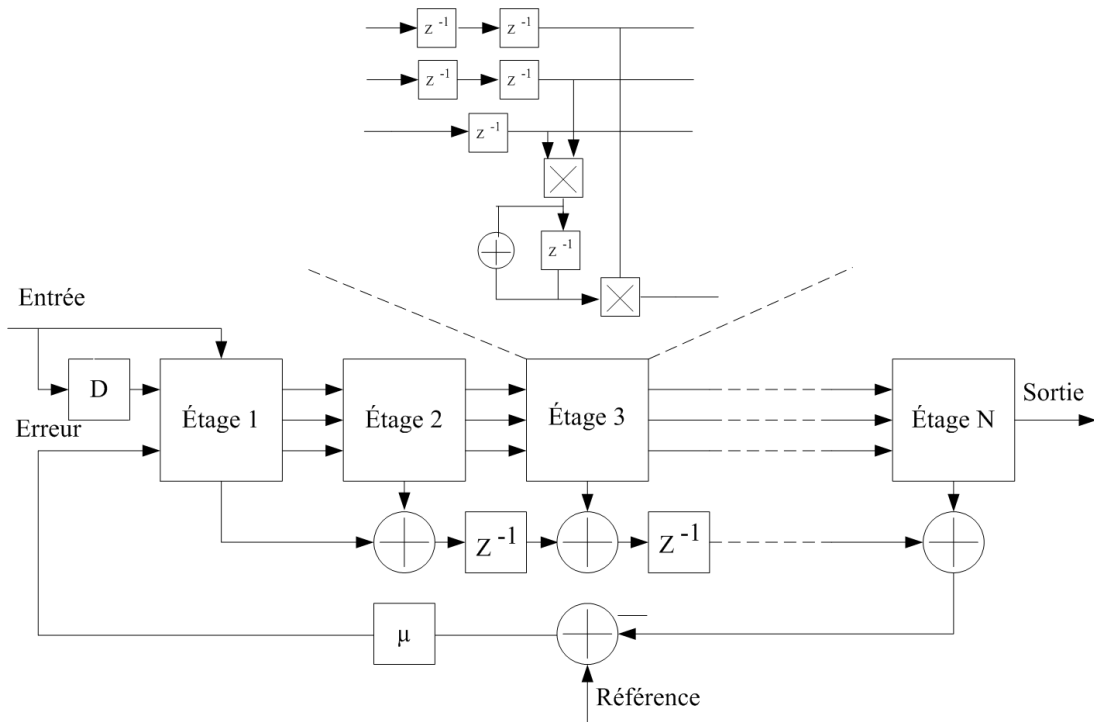


FIG. 5.28 – Le filtre NDLMS.

augmentation du débit du filtre. En plus, la grande différence avec le filtre LMS consiste, grâce au parallélisme, au fait que l'augmentation de l'ordre du filtre n'implique pas une diminution du débit. Ainsi pour avoir plus de précision au niveau de la sortie du filtre, nous pouvons augmenter son ordre, sans que cela diminue la rapidité du filtre comme dans le cas du filtre LMS.

Cependant, il a été montré dans [58] que le filtre DLMS présente certains inconvénients. Il s'agit surtout de l'augmentation du temps de la convergence du filtre et des restrictions sur le pas d'adaptation de l'algorithme pour assurer la stabilité. L'équation 5.7 devient

$$0 < \mu < \frac{2}{N\sigma_x^2} \sin\left(\frac{\pi}{2(2D+1)}\right) \quad (5.14)$$

En outre, pour les applications utilisant des signaux non stationnaires, le filtre DLMS est insuffisant. L'erreur de poursuite augmente dans des proportions plus importantes que celles dans le filtre LMS. Le filtre NDLMS (Normalized Delayed LMS) constitue la solution pour la convergence de l'algorithme quand il s'agit d'utiliser des signaux non stationnaire [67]. Le filtre NDLMS est géré par les mêmes équations que le filtre LMS. Seule l'équation d'adaptation des coefficients du filtre, 5.6, change de forme pour tenir compte de la dépendance du pas d'adaptation avec l'énergie du signal à adapter et des éléments des délais insérés :

$$H(n+1) = H(n) + \frac{\mu}{X^t(n+1)X(n+1)} X(n+1-D)e(n+1-D) \quad (5.15)$$

Pour D égale à 0, l'équation précédente est similaire à l'équation 5.13 et le filtre NDLMS a le même comportement que le filtre NLMS.

5.4.2.2 Validations

Pour des signaux stationnaires, le filtre NDLMS a le même rôle que les filtre LMS et NLMS. Les résultats de simulation sont donnés dans la figure 5.29. Les raies de disparités de l'erreur de gain et des erreurs d'ouverture sont réduites de 20 dB pour un filtre NDLMS d'ordre 8.

Les filtres adaptatifs normalisés ont l'avantage d'avoir un pas d'adaptation de coefficients dépendant de l'énergie du signal à filtrer. Ainsi, pour des fréquences proches de la fréquence de Nyquist, figure 5.30.a, ces filtres se comportent mieux que le filtre LMS.

Sur cette figure, avec ou sans correction, le *SNR* diminue en s'approchant de fréquence de Nyquist. Mais, pour les filtres normalisés le *SNR* présente de meilleures valeurs.

Conjointement à ce test fréquentiel, un test statistique a été établi. Nous faisons varier l'amplitude du signal d'entrée jusqu'à la pleine échelle, figure 5.30.b. En théorie, le *SFDR* diminue pour des amplitudes faibles ou proches de la pleine échelle de conversion. Le graphique nommé sans correction, sur la figure 5.30.b, est conforme avec cette théorie. En ajoutant une correction par le filtre LMS, le *SFDR* présente des valeurs très faibles, voire inférieures au cas sans correction.

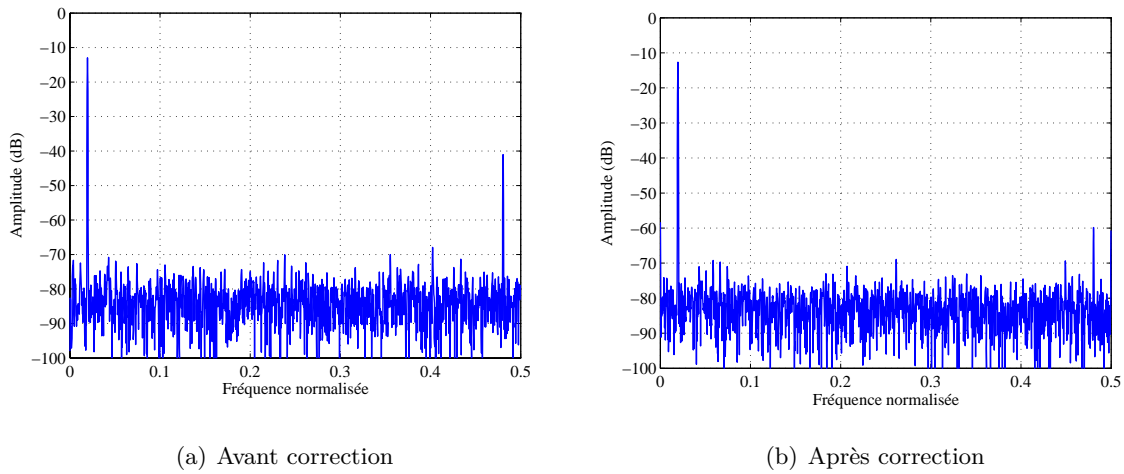
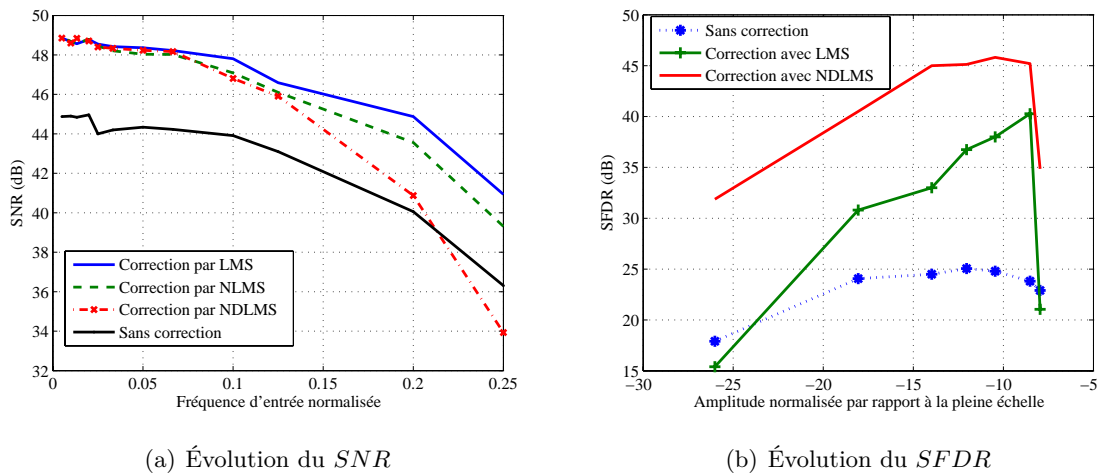


FIG. 5.29 – Spectres de sortie avant et après correction par filtre NDLMS

FIG. 5.30 – Évolution du SNR et du $SFDR$.

Ceci s'explique par le pas constant de l'adaptation des coefficients de ce filtre. En revanche, pour le filtre NDLMS, le $SFDR$ reste toujours supérieur et présente des pics d'amélioration de 22 dB.

5.5 Conclusion

Après avoir détaillé l'état de l'art des méthodes de correction des erreurs de disparités dans les TIADC, nous avons proposé dans ce chapitre de nouvelles méthodes de calibration numérique.

Dans un premier temps, une méthode en différé a été proposée. Elle se base sur des calculs mathématiques pour extraire les valeurs des erreurs de gain et d'offset. Une fois ces valeurs déterminées, elles sont extraites de la sortie globale du TIADC. Les résultats de simulation montrent l'amélioration des performances spectrales.

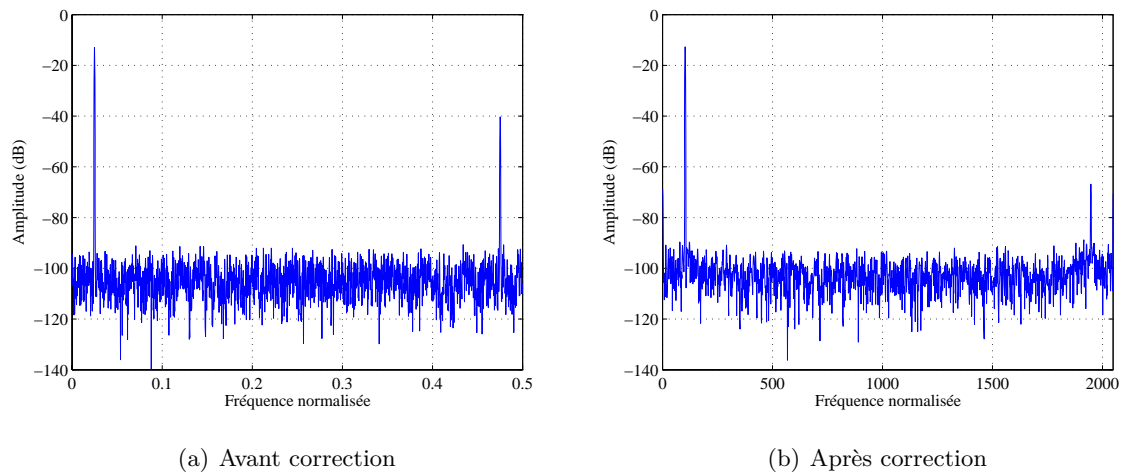


FIG. 5.31 – Spectres de sortie pour un TIADC de résolution 10 bits.

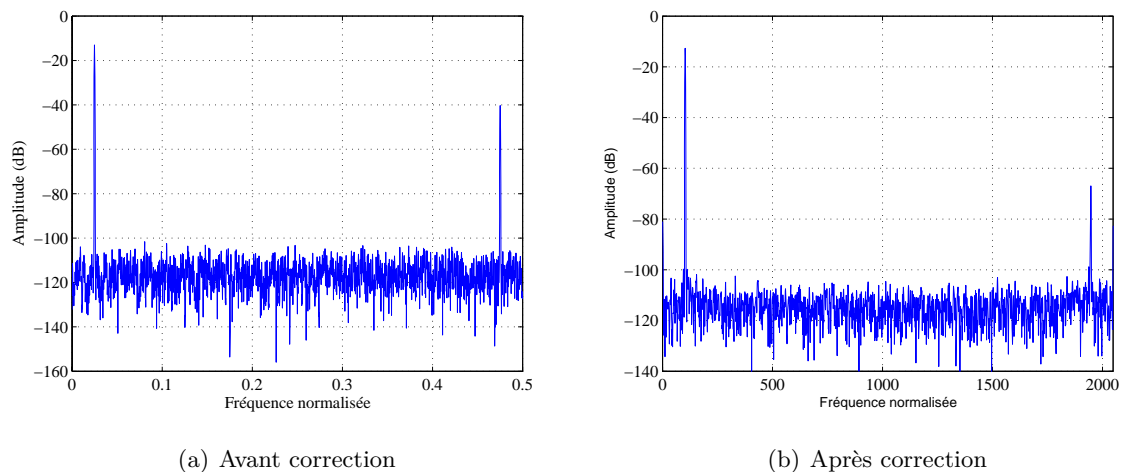


FIG. 5.32 – Spectres de sortie pour un TIADC de résolution 12 bits.

Ensuite, nous avons proposé une méthode en ligne adaptative qui constitue le futur des méthodes de correction des TIADC, [68]. Cette méthode utilise la notion de filtrage adaptatif et la notion d'interpolation numérique. Plusieurs filtres adaptatifs, basés sur la minimisation de l'erreur quadratique moyenne, ont été proposés. Le filtre LMS est sans doute le plus simple à implémenter,

mais tout changement d'amplitude apporté sur le signal d'entrée devrait être accompagné par un changement du pas d'adaptation des coefficients. Le cas des signaux non stationnaires limite l'application de ce type de filtre à pas d'adaptation constant. Le filtre NLMS prend en compte l'énergie du signal à filtrer et améliore les performances du filtre LMS.

Une autre amélioration a été apportée au filtre NLMS pour augmenter sa rapidité de fonctionnement. L'idée est de chercher un parallélisme dans la structure du filtre FIR pour séparer les différents calculs des coefficients et diminuer le temps de cycle. Cette rapidité est obtenue par insertion d'éléments de délais sur l'erreur du filtre NLMS. Le filtre résultant est le filtre NDLMS. Nous avons testé la solution à base de filtre NDLMS pour différentes résolutions du TIADC utilisant les mêmes caractéristiques du filtre adaptatif en terme d'ordre et de pas d'adaptation et les mêmes paramètres de simulation. Les résultats sont reportés dans les figures 5.31, 5.32 et 5.33. Au delà d'une résolution excédant les 12 bits et pour un retard d'ouverture supérieure ou égale à 10% de la période d'échantillonnage, les erreurs de disparités ne sont pas corrigées et la raie fréquentielle localisée à $\frac{f_s}{2} - f_0$ n'est pas réduite. Ce fait est considéré comme la limite des méthodes de correction des TIADC qui sont plus sensibles aux erreurs de retard d'ouverture pour des résolutions assez élevées.

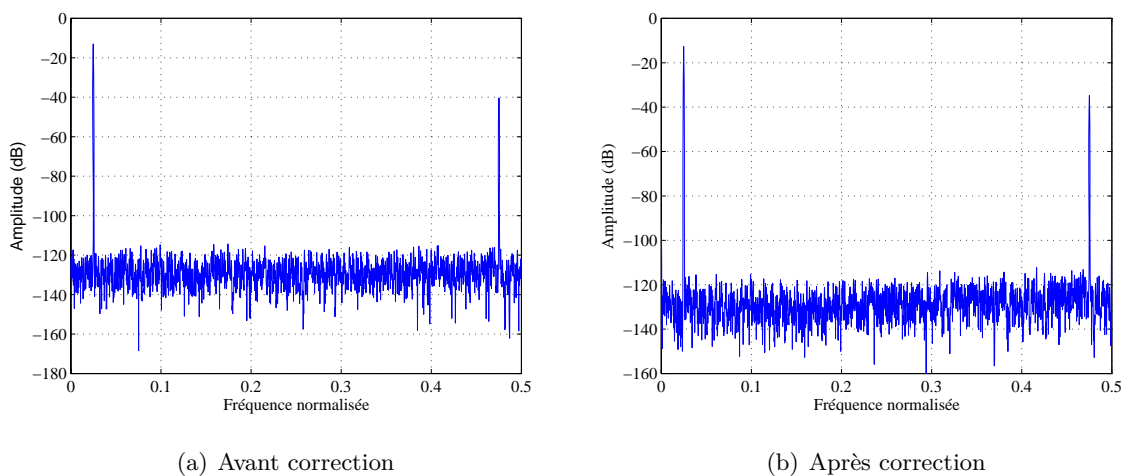


FIG. 5.33 – Spectres de sortie pour un TIADC de résolution 14 bits.

Enfin, la finalité de la description mixte des TIADC est de faire interagir la partie de post-correction numérique avec les modèles comportementaux des CAN dans la même interface de modélisation et de simulation. Le deuxième intérêt est de pouvoir réaliser une implémentation de la méthode de correction. Le dernier chapitre de ce manuscrit est consacré à l'implémentation sur FPGA de la solution proposée ainsi qu'aux tests et aux mesures.

Implémentations, tests et mesures

Sommaire

6.1	Introduction	131
6.2	La modélisation mixte (analogique-numérique)	132
6.2.1	Environnement de la modélisation mixte	132
6.2.2	La méthodologie de la modélisation mixte	134
6.2.3	Intérêt de la modélisation mixte	136
6.3	Résultats d'implémentation	139
6.3.1	Résultats de synthèse	139
6.3.2	Validation fonctionnelle	143
6.4	Tests et mesures	144
6.4.1	Validation de la méthode en différé avec des données expérimentales	144
6.4.2	Validation de la méthode en ligne par des mesures	146
6.5	Conclusion	150

6.1 Introduction

Le dernier chapitre présente les résultats d'implémentation, les différents tests et les mesures afin de valider les deux méthodes de correction proposées. Dans un premier temps, le choix de la méthodologie de modélisation pour décrire la structure du TIADC et les méthodes de correction proposées est justifié. Le TIADC est modélisé dans un environnement de simulation mixte (ADVancedMS) basé sur le langage VHDL-AMS tirant profit de la flexibilité des circuits

reconfigurables de type FPGA (Field Programmable Gate Array) pour l'implémentation.

Les résultats de l'implémentation matérielle sont ensuite donnés pour évaluer la qualité de la solution proposée. La méthode de correction proposée à base d'un filtre NDLMs et d'une interpolation en cosinus, a été implémentée sur plusieurs cibles programmables de générations différentes et de différents fabricants afin d'évaluer les propriétés et les caractéristiques du filtre sur plusieurs cibles. Enfin, dans le but d'avoir une analyse complète de ce qui a été proposé dans les chapitres précédents, deux bancs de tests sont mis en oeuvre afin de valider expérimentalement les résultats théoriques.

- Le premier est utilisé pour tester la méthode en différé. Un logiciel de caractérisation des CAN (ADCLab) développé au sein de l'équipe Circuit et Systèmes Numériques et Mixtes du laboratoire IMS est employé. La méthode de correction en différé est intégrée dans ce logiciel de caractérisation qui récupère les données de sorties d'un convertisseur entrelacé pour les traiter. Il s'agit donc d'une validation fonctionnelle basée sur l'utilisation de données expérimentales.
- Le deuxième est utilisé pour valider la méthode en ligne. La méthode en ligne proposée est implémentée dans un FPGA VirtexIIPro de la société Xilinx disponible au laboratoire et le logiciel de caractérisation des CAN est utilisé pour récupérer les données à la sortie de cet FPGA.

Les résultats de tests et de mesures sont donnés afin de valider les méthodes de correction des erreurs de disparité dans les TIADC.

6.2 La modélisation mixte (analogique-numérique)

La modélisation peut être définie par un *modèle* utilisant un *langage* de modélisation ainsi qu'un *simulateur*, [69]. Le modèle, le langage et le simulateur définissent l'environnement de la modélisation.

6.2.1 Environnement de la modélisation mixte

6.2.1.1 Modèle

Un modèle est une représentation abstraite d'un phénomène physique dont on ne conserve que les aspects essentiels à une certaine utilisation, [70]. Dans le cadre de nos travaux, nous nous intéressons aux modèles mathématiques afin de décrire le comportement des CAN et par la suite du TIADC utilisant les équations mathématiques détaillées dans les chapitre 2 et 3. Un modèle mathématique définit donc les fonctions, ou plus généralement le comportement, d'un système en exprimant des relations de causalité ou des équations définissant des relations entre les différentes variables du système. Les variables sont des grandeurs mathématiques représentant

des grandeurs physiques.

Il y a différentes façons pour modéliser le comportement d'un système.

- Le comportement continu : il est mathématiquement représenté par une fonction à valeurs réelles d'une variable réelle indépendante, généralement le temps ou la fréquence. Un tel comportement peut être représenté par des relations algébriques linéaires ou non linéaires, différentielles linéaires ou non linéaires.
- Le comportement discret : il est mathématiquement représenté par une équation aux différences. Les variables dépendantes sont à valeurs réelles, mais la variable indépendante est discrète (usuellement un multiple entier d'une unité de base).
- Le comportement logique : il est mathématiquement représenté par des équations booléennes. Les variables dépendantes ne peuvent prendre qu'un ensemble fini de valeurs quantifiées, ou états logiques (0,1,X, etc...).

Un système microélectronique peut combiner des circuits ayant des comportements de natures différentes liées à l'hétérogénéité des systèmes actuels. Pour simuler le fonctionnement d'un système, des analyses se font afin de valider un modèle. La nature du système impose le choix du type du simulateur utilisé.

6.2.1.2 Simulateur

Le simulateur est l'un des outils essentiels d'aide à la conception assistée par ordinateur. Il existe trois grandes familles de simulateurs.

- Les simulateurs numériques : les premiers simulateurs numériques sont apparus dans les années 1970. Leur principe est de déterminer le fonctionnement booléen des circuits numériques constitués de portes logiques. Ce sont des simulateurs événementiels c'est-à-dire qu'ils ne se réveillent que lorsqu'un événement se produit. Ceci leur permet d'être plus rapides que les simulateurs analogiques. Ils analysent les circuits uniquement dans le domaine temporel discret en manipulant uniquement deux variables : l'état logique et le délai qu'il subit. Parmi les simulateurs numériques on peut citer ModelSim de Mentor Graphics [71] ou NC-Sim de Cadence [72].
- Les simulateurs analogiques : ils permettent d'analyser les circuits électroniques au niveau transistor. Il existe différents types d'analyses : l'analyse DC qui détermine le point de fonctionnement statique, l'analyse transitoire qui détermine l'évolution temporelle des tensions et des courants, l'analyse AC qui est une analyse fréquentielle petit signaux ou l'analyse RF pour les circuits Radio Fréquence. Lors d'une analyse transitoire, et contrairement aux simulateurs numériques, les simulateurs analogiques analysent le circuit à chaque instant et non pas à chaque événement, ce qui les rend moins rapides. Par contre, leur précision est nettement

supérieure puisqu'ils utilisent des modèles non linéaires et complets de transistors. Les plus connus des simulateurs analogiques sont SPICE de MicroSim [73], Eldo de Mentor Graphics [74] ou Spectre de Cadence [75].

- Les simulateurs mixtes : Avec le besoin de plus en plus important de concevoir des circuits mixtes, diverses générations de simulateurs mixtes se sont succédées. La dernière génération est basée sur un coeur de simulation unique. Ces outils intègrent tous les algorithmes nécessaires pour simuler les parties analogiques et numériques directement dans le même programme ; on n'a donc plus besoin de co-simulation. Cette évolution a été rendu possible par l'apparition du standard VHDL-AMS (Very high speed integrated circuit Hardware Description Language-Analog and Mixed Signal), langage de description qui permet de modéliser les parties analogiques et les parties numériques dans le même environnement de simulation. Mentor Graphics propose l'environnement de simulation mixte ADVanceMS, [76].

6.2.1.3 Langage

Deux grandes familles de langages existent dans le domaine de la conception.

- Les langages de programmation : ces langages tels que Java, C/C++, FORTRAN sont utilisés souvent pour vérifier certaines fonctionnalités ou programmer des mini-simulateurs. Ces langages se caractérisent par leur souplesse de leur mise en oeuvre dans certaines applications numériques.
- Les langages de description : ce sont des langages utilisés pour décrire les éléments matériels et le comportement d'un système électronique. Il existe plusieurs langages normalisés pour la description RTL et la synthèse logique dont nous citons : SystemC, VHDL (Very high speed integrated circuit Hardware Description Language), très utilisé en Europe et Verilog . D'autres versions normalisée de ces deux derniers langages sont apparues pour la modélisation analogique (VHDL-A et Verilog-A) et pour la modélisation mixte (VHDL-AMS et Verilog-AMS).

6.2.2 La méthodologie de la modélisation mixte

La méthodologie descendante (TOP-DOWN) employée pour valider les travaux détaillés dans ce manuscrit est présentée dans la figure 6.1. Tout d'abord, une vérification comportementale de la structure du TIADC et des algorithmes de correction des erreurs de disparité a été réalisée à l'aide de Matlab utilisant des bibliothèques spécialisées notamment en traitement de signal. Ensuite, une description structurelle est développée à l'aide de l'outil Simulink. Le but de ces deux premières étapes est de valider le fonctionnement de la méthode de correction, de paramétrer les algorithmes employés et d'évaluer leurs performances.

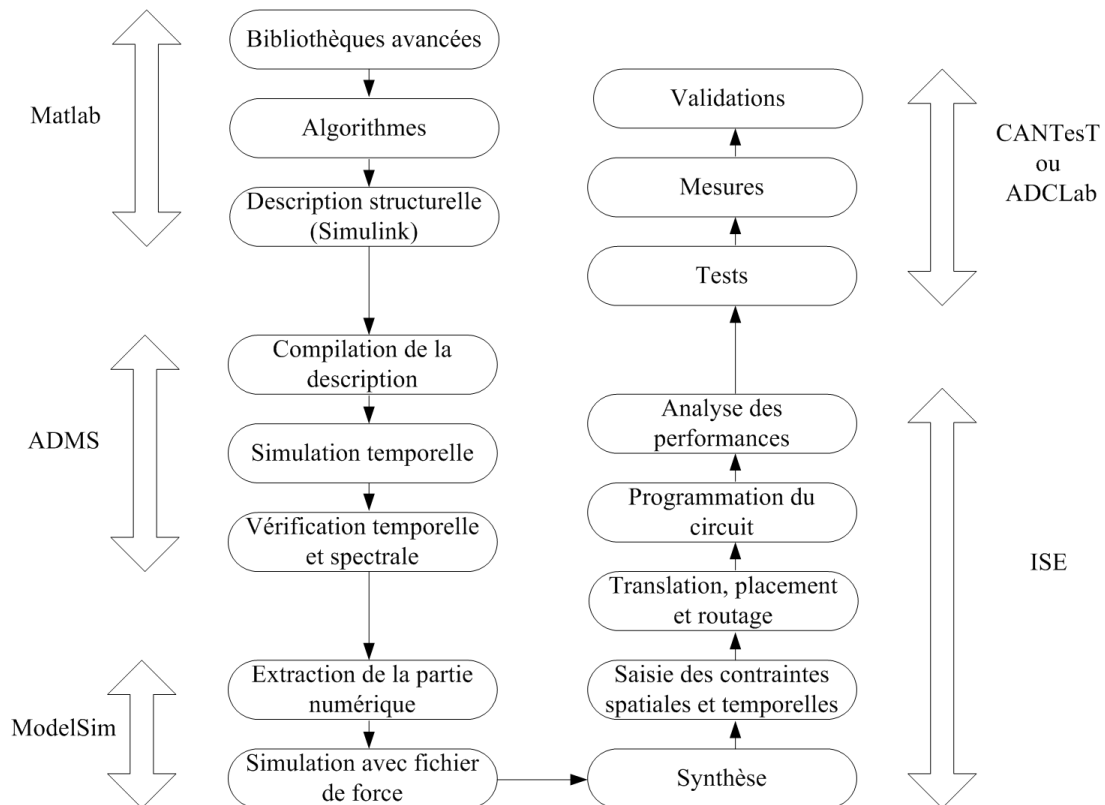


FIG. 6.1 – Méthodologie de conception.

Dans le but d'obtenir une description matérielle, deux alternatives se présentent : la simulation mixte et la co-simulation. Dans l'annexe A, une étude comparative de ces deux approches est présentée. L'intérêt de la simulation mixte est mis en évidence pour deux raisons principales :

- La simulation mixte utilise un seul environnement de simulation pour les parties analogiques et les parties numériques. Le signal analogique généré pour être employé à l'entrée du TIADC utilise la variable "`<NOW>`" pour remplacer la variable du temps "`<t>`". Cela va à l'encontre de la co-simulation qui, pour modéliser la partie analogique, utilise des signaux discrets avec un pas temporel suffisamment faible. Les deux méthodes ont été testées pour évaluer le jitter dans un échantillonneur bloqueur et les résultats de comparaison détaillés dans l'annexe A montrent que la simulation mixte offre une meilleure précision.
- La description VHDL générée lors de la co-simulation est liée aux outils utilisés : l'utilisation de SystemGenerator par exemple impose une implémentation sur des cibles matérielles de Xilinx. En outre, l'implémentation sur un ASIC ne peut pas se faire vu que la netlist générée utilise des bibliothèques propres à Xilinx. L'aspect standard et portabilité du langage VHDL ne sont donc pas conservés.

Ainsi, AdvancedMS est utilisé en tant que simulateur mixte. A ce niveau, la description structurale et comportementale établie à l'aide de Matlab est développée avec le langage VHDL-AMS. Des simulations temporelles et des analyses spectrales sont réalisées pour valider le fonctionnement globale de la structure du TIADC avec la méthode de correction.

Ensuite, la partie numérique (qui correspond à la méthode de correction) est extraite et simulée avec ModelSim utilisant un fichier de force pour s'assurer de la correcte fonction du système.

Les fichiers VHDL obtenues sont synthétisés avec l'outil ISE de Xilinx. Ensuite, le flot de conception de Xilinx pour les FPGA est utilisé (saisie de contraintes spatiales et temporelles, translation, mapping, placement, routage et programmation du circuit) pour déduire les performances du circuit (vitesse maximale de fonctionnement, occupation de surface silicium et consommation de puissance).

Enfin, des tests et des mesures sont réalisés afin de valider la méthodologie de modélisation et la méthode de correction. Pour ce faire, nous utilisons l'outil CANTesT pour la caractérisation des CAN dont la version de base est développée au sein du laboratoire IMS dans [77]. Nous avons adapter cette version pour assurer la caractérisation des CAN rapides. De nouveaux modes de fonctionnement (test en mode démultiplexé, test en mode entrelacé) sont ajoutés ainsi que de nouvelles fonctionnalités notamment celles qui concernent le calcul des paramètres spectraux afin d'estimer les contributions du filtre passe bande utilisé. La nouvelle version de CANTesT permet l'utilisation de l'analyseur logique HP16702 mené des sondes d'acquisitions rapides de type HP16760. L'outil CANTesT est développé en langage C dans un environnement Mac. Nous disposons aussi de la version ADCLab développée dans un environnement Windows avec Matlab.

6.2.3 Intérêt de la modélisation mixte

La modélisation mixte tire profit de la simulation mixte (Annexe A) utilisant le langage VHDL-AMS et du développement des cibles matérielles programmables, [78].

6.2.3.1 Apport de VHDL-AMS

Le langage VHDL-AMS est un sur-ensemble de VHDL. Cela signifie que toute description légale en VHDL l'est aussi en VHDL-AMS et produit par conséquence les mêmes résultats de simulation. Un modèle VHDL-AMS est constitué de deux parties principales : la spécification d'**entité** qui correspond à la vue externe du modèle et l'**architecture** de l'entité qui est la vue interne. Mais avant la spécification de l'entité, on fait appel aux **bibliothèques** utiles pour décrire l'architecture en précisant le contenu à exporter. Ces bibliothèques contiennent des fonctions prédéfinies telles que des fonctions arithmétiques, des fonctions mathématiques, des constantes physiques, thermiques ou électromagnétiques.

L'entité permet de définir les paramètres génériques et les entrées-sorties du modèle, à travers lesquels il communique avec son environnement. Les différentes informations pouvant être échangées sont supportées par des **ports** de type **signal** pour une information à événement discret, **quantity** pour des informations analogiques orientées et **terminal** pour les informations analogiques de type noeuds de connexion.

L'architecture est constituée d'une zone de déclaration et d'un corps dans lequel on définit le fonctionnement du modèle par l'intermédiaire d'instructions concurrentes, simultanées ou séquentielles. Dans un modèle VHDL-AMS, toutes les instructions peuvent cohabiter offrant ainsi la possibilité d'écrire des modèles pour des circuits analogiques et mixtes avec plusieurs niveaux d'abstraction. Pour une même entité, on peut également écrire plusieurs architectures.

L'un des avantages majeurs du VHDL-AMS réside dans le fait qu'il supporte des simulations mixtes : grâce à l'instruction **Break** il y a séparation et discontinuité entre la simulation à temps continu et discret. Ce langage fait de la modélisation comportementale des systèmes à temps continu et discret. On peut même exprimer le comportement dynamique de certaines quantités en utilisant les opérateurs **'dot** et **'integ** pour exprimer respectivement une dérivation ou une intégration. Enfin, VHDL-AMS est un langage transparent : il ne comprend pas de modèles pré-définis. Le concepteur possède la flexibilité de modéliser ses propres systèmes comportementaux et la possibilité de modifier ces modèles pour les adapter à ses besoins.

6.2.3.2 Cible matérielle

Dans cette section et en se basant sur les travaux de [79], nous montrons l'intérêt d'utiliser des circuits reconfigurables pour compléter la méthodologie de conception et pour implémenter la solution proposée.

Evolution des FPGA : Les circuits reconfigurables peuvent être classés sur la base de la granularité des ressources de traitement. La séparation des architectures se fait suivant trois branches : grain fin (ressources de traitement de type LUT), gros grain (ressources de traitement de type ALU) enfin multi-grains (mélangeant coeurs de processeurs, matrices d'ALU et/ou de LUT). Les FPGA (Field Programmable Gate array) sont des architectures reconfigurables à grain fin dont le concept a vu le jour aux années 80.

Les FPGA ont été longtemps utilisés uniquement pour le prototypage d'ASIC ou pour réaliser quelques parties de logique combinatoire dans un système, les FPGA ont tardé à trouver une véritable place dans le monde de l'électronique et ceci pour deux raisons. Tout d'abord ils ne bénéficiaient pas à leurs débuts de technologies assez avancées pour leur assurer une intégration à

la hauteur des besoins applicatifs. Mais aussi, les outils de placement-routage et de synthèse qui leur étaient dédiés n'étaient pas assez performants. Ceci s'explique par le fait que les FPGA ont à leurs débuts hérité des méthodologies de conception propres aux ASIC qui sont peu adaptées aux ressources de routage reconfigurables à grain fin. Les ressources de routage sont en effet la source principale de dégradation des performances d'une application entre une implémentation sur ASIC et sur FPGA. Aujourd'hui cela reste vrai et la contribution des ressources de routage s'accroît même à cause de la réduction des technologies.

Dans un contexte économique mondial incertain les FPGA apparaissent donc comme une solution flexible bien adaptée aux contraintes économiques telles que le temps de mise sur le marché et le potentiel d'évolution ou de flexibilité des produits. De plus, le modèle économique lié aux FPGA, qui est un modèle linéaire devient de plus en plus avantageux par rapport aux modèles ASIC dont le coût du premier échantillon fabriqué rend l'amortissement d'une telle solution long et viable économiquement uniquement pour de très grandes productions. Dans [80] l'auteur explique bien ce phénomène et montre que le nombre de circuits fabriqués à partir duquel la solution ASIC est économiquement plus rentable (point de cross-over) tend à augmenter avec l'évolution des technologies. Les solutions FPGA sont donc de plus en plus intéressantes tant sur le point technique qu'économique.

Structure des FPGA : Dans le cas des FPGA développés par les sociétés Xilinx et Altera, et dans la plupart des cas, l'élément configurable de base se compose d'une LUT à 4 entrées, d'une chaîne de propagation rapide de la retenue et d'un registre de sortie afin d'assurer la synchronisation des signaux comme on le voit sur la figure 6.2. Le nombre d'entrées des LUT n'est pas dû au hasard, des études montrent qu'il s'agit là d'un bon compromis entre les performances du circuit et les contraintes des algorithmes de placement-routage, leur complexité et leur efficacité, [81]. La plupart des FPGA, étant donnée leur taille, ont besoin de réseaux de distribution de l'horloge spécialement adaptés à une transmission homogène des horloges sur toute l'architecture. Pour les autres entrées/sorties, les blocs dédiés à la communication avec l'environnement extérieur sont aussi configurables afin de s'adapter aux différents standards de communication de l'environnement du circuit.

De plus, nombreuses sont les applications qui nécessitent la synthèse d'opérateurs du type multiplicateur, additionneur et multiplieur/accumulateur. S'il est possible, grâce aux chaînes de propagation rapide de la retenue de réaliser sur un petit nombre de LUT des additionneurs efficaces, ce n'est pas le cas pour les multiplieurs très coûteux en ressources. Les industriels ont donc choisi d'implanter de façon matérielle des multiplieurs reconfigurables (la reconfiguration intervient en particulier sur la taille des données à traiter) au sein même de la matrice de grain fin. En positionnant ces multiplieurs près des colonnes d'éléments mémoires et d'éléments reconfigurables

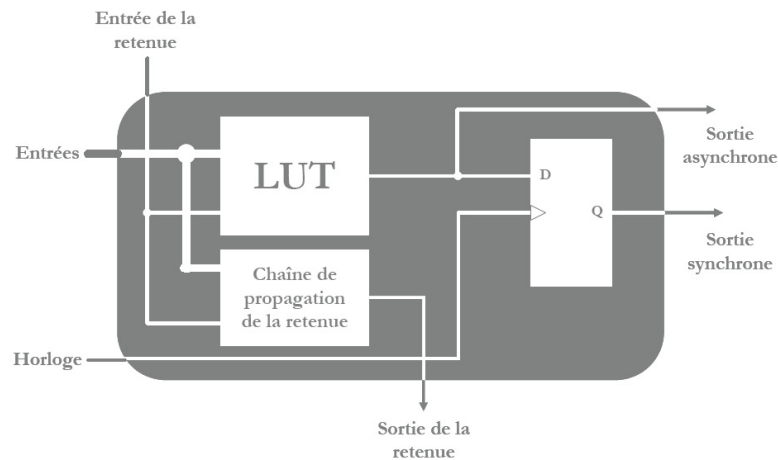


FIG. 6.2 – Élément configurable de base des FPGA classiques.

de grain fin, il est possible de synthétiser des opérateurs MAC. Cette solution fut retenue par Xilinx pour les composants Virtex-II, [82].

6.3 Résultats d'implémentation

Le processus d'implémentation comporte plusieurs étapes. La présente section détaille les résultats d'implémentation par étape. L'implémentation est réalisée par l'outil ISE Foundation de Xilinx. Cet outil comprend plusieurs logiciels pour les différentes étapes de l'implémentation. Tous ces outils sont utilisés sauf celui de l'analyse de la puissance (XPower). En effet, l'analyse de la puissance se fait d'une façon très approximative et par conséquent elle n'est citée que dans le cas où la différence entre plusieurs design est très importante.

6.3.1 Résultats de synthèse

L'opération de synthèse consiste à traduire le fichier VHDL au niveau RTL (Register Transfer Level). Le fichier analysé est ensuite transformé en une netlist. Le synthétiseur évalue donc les ressources consommées du circuit à synthétiser. Cela dépend évidemment des options de synthèse et du type de la carte cible.

6.3.1.1 Choix de l'option de synthèse

Les options de synthèse définissent les efforts à effectuer par le synthétiseur. Ce dernier peut privilégier la rapidité par rapport à la consommation de surface ou inversement.

Le FPGA choisie est un VIRTEX 2PRO XC2VP30 disponible au laboratoire. Il contient 13969

slices, une mémoire distribuée de 428 Kbit, des blocs RAM de 2448 Kbit, 136 multiplieurs sur 18 bits et deux PowerPC RISC Cores.

La solution en ligne proposée pour la correction des erreurs dans un TIADC composée du filtre NDLMs et de la méthode d'interpolation en cosinus est synthétisée par l'outil XST de Xilinx. Le tableau 6.1 illustre un exemple de synthèse avec deux options différentes. Une première synthèse est faite utilisant des multiplieurs sous forme de LUT (Look Up Table). La deuxième emploie des multiplieurs embarqués dans le FPGA. La solution proposée consomme peu d'éléments

	Slices	Flip Flop	LUT	MULT	IOB	GCLK	Fréquence
MULT en LUT	1346	630	2156	–	27	1	71.505MHz
Pourcentage	9 %	2 %	7 %	–	4 %	6 %	–
MULT embarqués	354	623	332	16	27	1	114.136MHz
Pourcentage	2 %	2 %	1 %	11 %	4 %	6 %	–

TAB. 6.1 – Résultats de synthèse avec deux options différentes.

logiques disponibles pour les deux options de synthèse. Dans la méthode proposée, le nombre de multiplieurs utilisés est égale à 16. Ce sont les multiplieurs du filtre FIR d'ordre 8. La première méthode de synthèse transforme ces multiplieurs utilisant des LUT à 4 entrées, des slices et des bascules. La deuxième, fait la correspondance entre ces multiplieurs et les blocs de multiplieurs embarqués. Elle utilise par conséquence moins de slices, de bascules et de LUT. Rappelons qu'il s'agit d'une multiplication de type $A \times B$ fournie par les bibliothèques standards de l'outil de synthèse logique. Des algorithmes de multiplication différents de type additions successives ou additions et décalages ont été testés. Les résultats de synthèse obtenus sont comparables à ceux d'une multiplication de type $A \times B$ au niveau ressources consommés, mais pour la fréquence maximale de fonctionnement elles sont bien en dessous.

En effet pour une multiplication de type additions successives, le temps nécessaire pour la multiplication de deux vecteurs de 8 bits utilisés dans le filtre FIR peut atteindre 64 cycles d'horloges. La fréquence maximale de fonctionnement est donc divisée par 64. Dans le cas d'une multiplication par additions et décalages, le temps de multiplication est de 8 cycles d'horloges et la fréquence maximale est divisée par 8.

6.3.1.2 Choix des cibles FPGA

Les résultats de la synthèse logique dépendent des options utilisées ainsi que de la cible choisie. En effet, les différents FPGA n'ont pas été conçus avec la même technologie et par conséquent les résultats issus de la synthèse logique sont sensiblement différents. En général, le FPGA est

d'abord choisi pour être adapté, le mieux possible, à ce qu'on souhaite synthétiser. Le nombre d'entrées/sorties IOB (Input Output Block) du FPGA doit obligatoirement être supérieur aux entrées/sorties du design. La fréquence maximale fournie par le FPGA doit être supérieure ou égale à celle requise par le programme qu'on souhaite concevoir. Ensuite, le choix peut être orienté par le coût du FPGA, par ses performances ou par un compromis entre ces deux critères de choix. Nous exposons dans cette section les différents résultats de synthèse logique obtenus pour différentes cibles architecturales.

Pour couvrir une large plage de FPGA, nous orientons d'abord la synthèse sur deux familles de FPGA : les FPGA Xilinx et les FPGA Altera. Ces deux constructeurs représentent environ 80% des cibles disponibles sur le marché. Ensuite, nous nous intéressons aux FPGA de différentes générations : les FPGA d'anciennes générations et les FPGA de nouvelles générations.¹ La synthèse se fait par l'outil QuartusII pour les FPGA de la famille Altera. Nous présentons dans le tableau 6.2 les différents résultats de synthèse que nous avons classé par génération :

- les résultats de synthèse pour des FPGA d'anciennes générations Xilinx (Spartan 2, Spartans 3, Spartan2E, Spartan3A et Virtex2) et Altera (Cyclone, Stratix) sont présentés dans la première partie du tableaux 6.2. Les multiplieurs utilisés pour la synthèse sont en LUT vu que certains FPGA ne disposent pas de multiplieurs embarqués. La synthèse pour un FPGA Cyclone utilise plus d'éléments logiques et de fonctions combinatoires que la synthèse pour les Spartan. Les estimations de la fréquence maximale de fonctionnement sont sensiblement identiques.
- les résultats de synthèse pour des FPGA de nouvelles générations Xilinx (Spartan 3E, Spartans 3L, Virtex4, Virtex5) et Altera (Stratix2) sont présentés dans la deuxième partie du tableaux 6.2. Les résultats de synthèse utilisant des FPGA de nouvelle génération montrent une occupation de surface silicium et des éléments logiques inférieure aux résultats présentés dans le cas des FPGA d'anciennes générations. Ceci est dû au fait que plusieurs fonctions sont associées aux blocs DSP. D'un autre coté, la fréquence maximale de fonctionnement augmente par rapport à celle obtenue pour les anciens FPGA. En effet, les FPGA de nouvelle génération sont conçus avec des technologies plus développées.

Parallèlement à l'étude des résultats de synthèse sur différentes familles de FPGA, une étude comparative entre le filtre LMS et DLMS a été menée pour valider la théorie développée dans le chapitre précédent. Il s'agit de comparer sur une cible (ViretexII PRO) la variation de la fréquence d'horloge maximale atteinte par la méthode de correction en fonction de l'ordre du

¹Les FPGA de nouvelles générations ont permis l'élaboration de nouvelles applications tirant parti de leurs caractéristiques propres tels qu'un fort parallélisme matériel et des possibilités de reconfiguration statique et/ou dynamique. Les FPGA de nouvelle génération ont intégré des unités arithmétiques spécialisées pour le traitement de signal.

Famille	Fabriquant	Circuit	Année	Flip Flop	LUT	DSP	Fréquence
Spartan2E	Xilinx	XC2S600	2002	626	2572	–	52.45MHz
Spartan3A	Xilinx	XC3S1400	2002	630	2156	–	54.28MHz
Spartan2	Xilinx	XC2S200	2000	626	2572	–	46.283MHz
Spartan3	Xilinx	XC3S4000	2002	630	2156	–	49.461MHz
Virtex2	Xilinx	XC2V8000	2001	630	2156	–	61.41MHz
Virtex2PRO	Xilinx	XC2VP30	2002	630	2157	–	71.505MHz
Virtex2Milit	Xilinx	XQV2000	2002	626	2572	–	53.41MHz
Stratix	Altera	EP1S10F	2002	630	296	–	–
Cyclone	Altera	EP1C4F324	2002	630	2720	–	–
Spartan3E	Xilinx	XC3S1600	2005	629	162	16	87.67MHz
Spartan3L	Xilinx	XC3S1000L	2004	629	162	16	85.22MHz
Virtex4	Xilinx	XC4VFX140	2005	627	146	16	142.41MHz
Virtex5	Xilinx	XC5VLX330t	2006	–	162	16	177.73MHz
Stratix2	Altera	EP1C4F324	2005	630	298	24	–

TAB. 6.2 – Résultats de synthèse pour différentes génération de FPGA.

filtre FIR utilisé. Les résultats reportés sur la figure 6.3 montrent que pour un ordre unitaire du filtre FIR (un seul PM), les fréquences maximales de fonctionnement sont identiques. Au delà de l'ordre 1, les deux filtres adaptatifs présentent un comportement différents : la fréquence maximale de fonctionnement décroît continuellement avec l'ordre du filtre FIR pour l'algorithme LMS alors que le parallélisme de la structure du filtre DLMS rend la fréquence maximale de fonctionnement constante. A titre d'exemple, pour un ordre de filtre égale à 6, la fréquence maximale de fonctionnement de la méthode de correction par LMS est égale à 62,4 MHz alors que pour le même ordre du filtre, la méthode de correction par DLMS présente une fréquence maximale de fonctionnement égale à 118,2 MHz. Cependant, l'augmentation de la fréquence de fonctionnement est suivie par une légère augmentation de la surface consommée puisque le filtre DLMS utilise des éléments de retards afin d'assurer le parallélisme de la structure. Le compromis vitesse maximale de fonctionnement et surface consommée reste en faveur de la solution de correction utilisant le filtre DLMS.

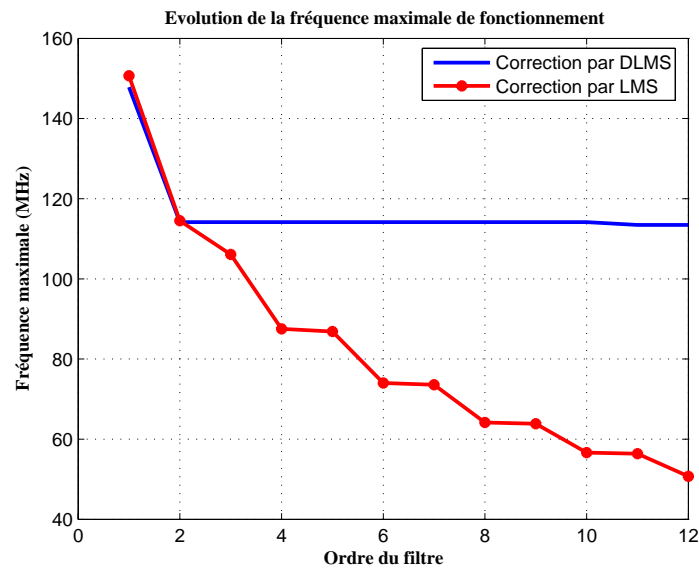


FIG. 6.3 – Évolution de la fréquence maximale de fonctionnement après synthèse.

6.3.2 Validation fonctionnelle

Nous proposons dans cette section d'étudier le comportement de la méthode de correction en ligne sur un FPGA. Pour s'assurer du bon fonctionnement de la méthode de correction implémentée dans le FPGA, nous proposons d'intégrer, en plus du composant de correction, un processeur (picoblaze) dans le FPGA VIRTEX-2 PRO pour agir en tant qu'interface d'entrée/sortie conformément à la figure 6.4.

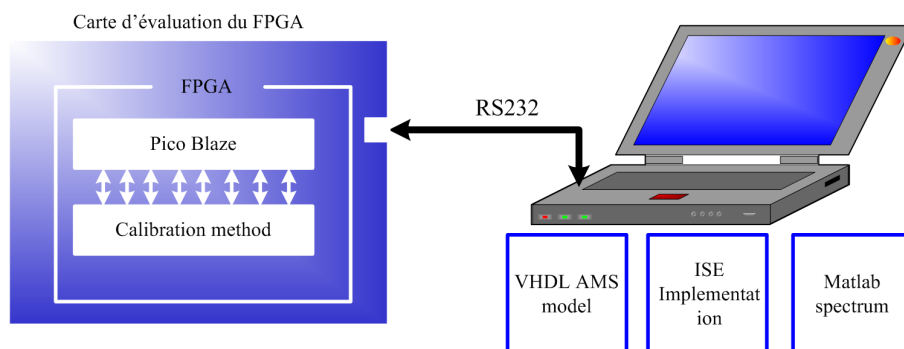


FIG. 6.4 – Schéma bloc de test de la méthode de correction implémenté sur FPGA.

Les entrées du processeur sont le vecteur de sortie du modèle du convertisseur entrelacé, l'horloge et la variable d'initiation de la correction. En sortie, nous récupérons la réponse de la méthode de correction dont le spectre est montré dans la figure 6.5. Nous choisissons, comme dans les

chapitres précédents, une résolution du CAN égale à 8 bit, par conséquent, la taille du vecteur d'entrée sortie des données au processeur est égale à 8 bits. Les erreurs d'offset et de gain

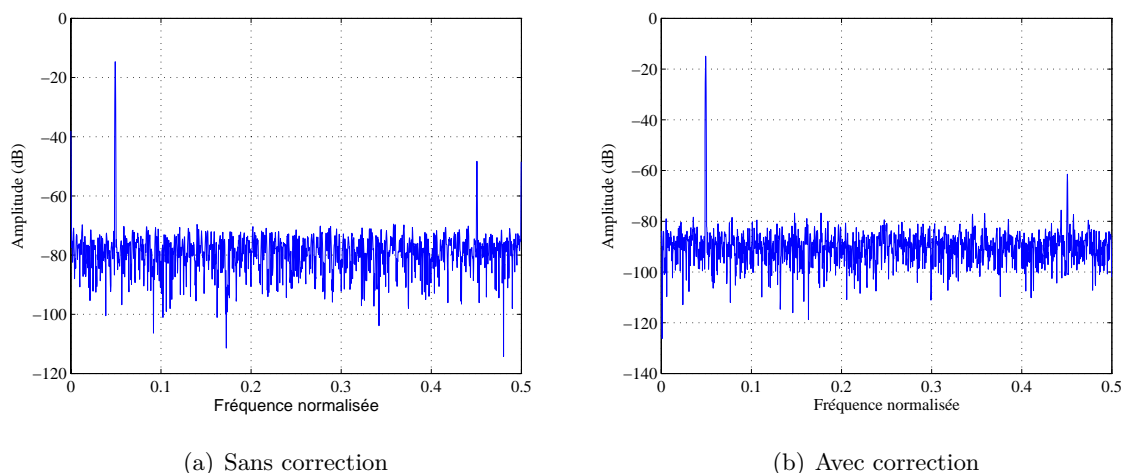


FIG. 6.5 – Spectre de sortie du TIADC avant et après implémentation de la méthode de correction.

sont éliminées suite à la correction, par contre celles du retard d'ouverture sont réduites ce qui explique la diminution de l'amplitude de la raie localisée à $\frac{f_s}{2} - f_0$. L'algorithme proposé est donc validé par simulation dans un environnement mixte et ensuite expérimentalement sur FPGA pour s'assurer des performances du système sur une cible matérielle. La dernière partie est consacrée aux mesures pour valider la méthode proposée dans des conditions de fonctionnement réelles.

6.4 Tests et mesures

Deux configurations de test sont réalisées. La première permet d'extraire et de mémoriser les données en sortie du TIADC sous calibration et d'appliquer la méthode de correction en différé intégrée dans le logiciel de caractérisation des CAN ADCLab. La deuxième configuration concerne la méthode en ligne : la sortie du TIADC est liée à un FPGA dans lequel la méthode de correction en ligne est implantée. Ensuite, le vecteur de sortie du FPGA est traité dans ADCLab pour calculer les paramètres spectraux avant et après la compensation.

6.4.1 Validation de la méthode en différé avec des données expérimentales

La méthode de correction en différé est une méthode de compensation logicielle développée sous Matlab. Par conséquent le test de cette méthode se fait en différé. Dans cette section, nous

validons les algorithmes de correction intégrés dans ADCLab à l'aide de données expérimentales. Une photo du banc de tests mis en oeuvre est donnée dans la figure 6.6. Le banc de tests

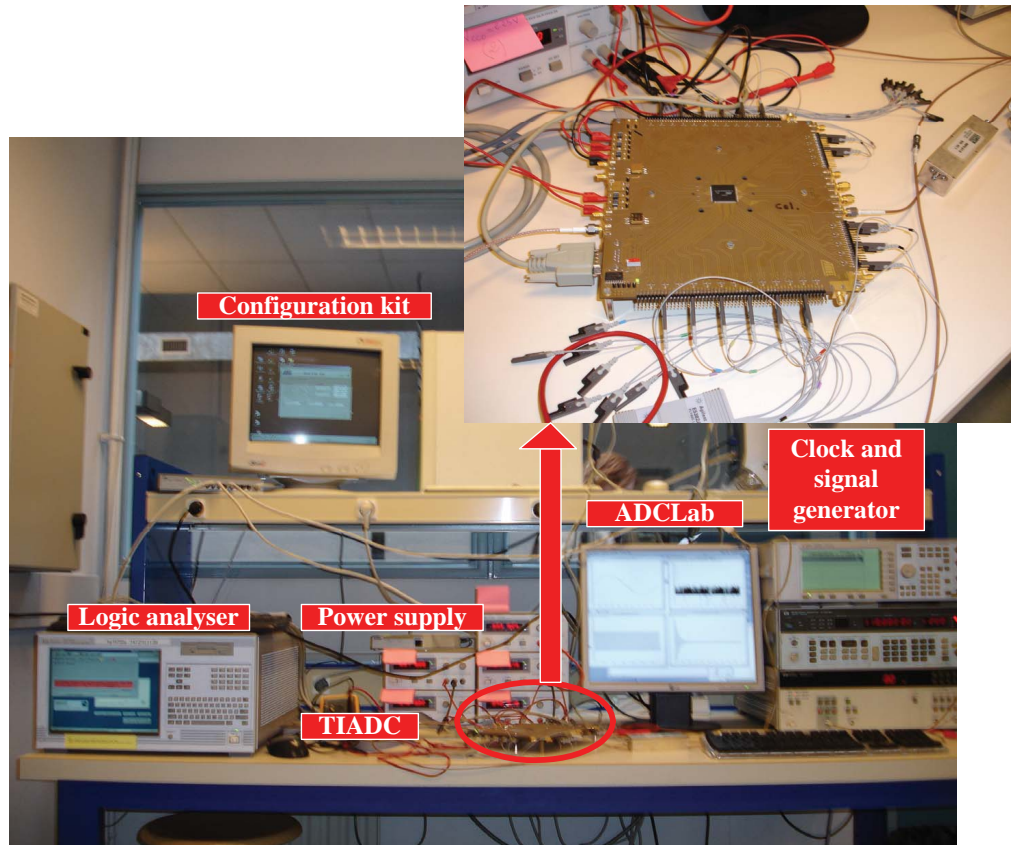


FIG. 6.6 – Banc de test.

comprend :

- un générateur E4400B de Agilent fonctionnant entre 250 kHz et 1 GHz pour le signal d'entrée du convertisseur ;
- un générateur HP8656B de Hewlett Packard fonctionnant entre 100 kHz et 990 MHz pour le signal d'horloge du convertisseur ;
- un filtre du signal d'entrée dont la bande passante est centrée sur la fréquence d'entrée afin d'améliorer les performances spectrales du signal d'entrer ;
- le circuit de convertisseur entrelacé sous calibration : AD8400D de résolution 8 bits de Atmel ;
- le logiciel de configuration du TIADC permettant l'ajustement des retards d'ouvertures et le choix de la configuration (un seul CAN, CAN entrelacés, horloge du premier CAN complétementé...);
- une liaison RS232 entre le convertisseur et le logiciel de configuration ;
- cinq boîtiers d'alimentations pour le convertisseur et la carte d'évaluation du convertisseur ;

- un analyseur logique HP16702B de Hewlett Packard munis des cartes d’acquisition HP16760 pour recevoir les données convertis par le TIADC ;
- le logiciel ADCLab de caractérisation des CAN communicant avec l’analyseur logique à l’aide du protocole TCP/IP.

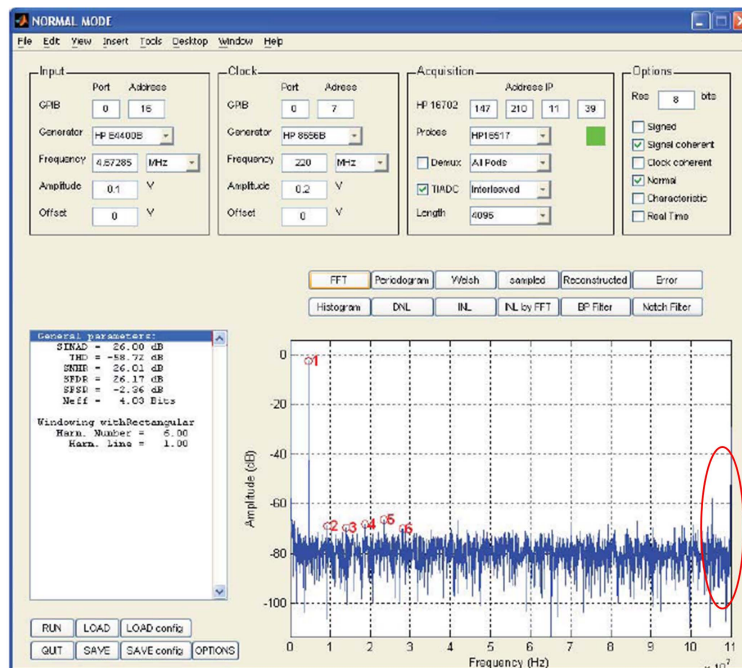
Le logiciel ADCLab permet de récupérer les données à la sortie des convertisseurs. Ensuite, des analyses spectrales (FFT et paramètres spectraux) et statistiques (histogramme, DNL et INL) sont mises en oeuvre pour évaluer la sortie. La figure 6.7 montre les spectres de la sortie avant et après l’application de la correction. Les raies de disparités des erreurs d’offset et de gain sont éliminées. Pour une acquisition sur 4096 points, le SNR passe de 4.02 bits à 6.36 bits et le $SFDR$ passe de 26.17 dB à 50.14 dB pour une fréquence d’échantillonnage de 220 MHz et un signal d’entrée cohérent à 5 MHz.

L’amélioration importante du $SFDR$ s’explique par la qualité de la méthode de correction appliquée et mais aussi par la précision du signal d’horloge appliqué et par le fait que le logiciel de configuration fourni avec la carte, propose une calibration du retard d’ouverture. Avant d’appliquer la méthode de correction, la sortie du TIADC présente uniquement des erreurs d’offset et de gain qui sont par la suite corrigées par la méthode en différé proposée.

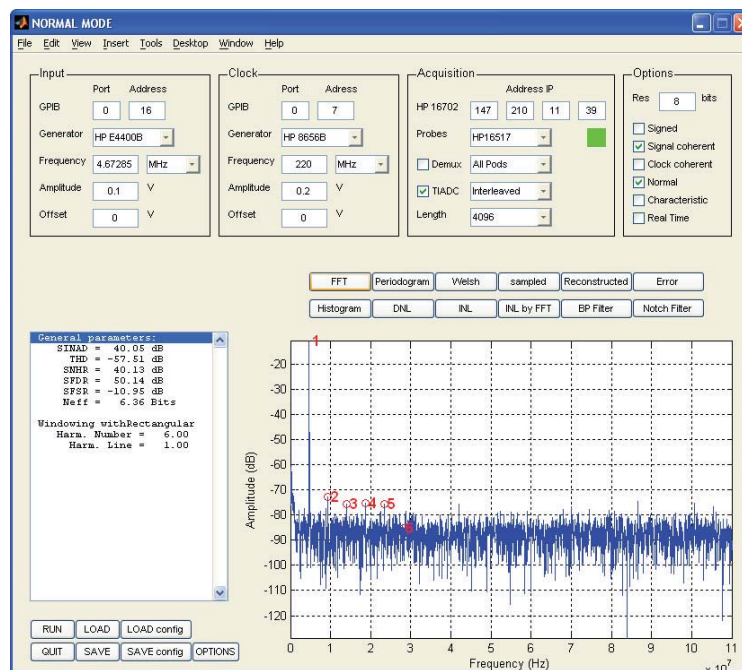
Le même banc de test a servi pour valider la méthode en ligne qui a été intégrée dans ADCLab. Les résultats de validation par des données expérimentales prouvent le bon fonctionnement de la méthode de correction.

6.4.2 Validation de la méthode en ligne par des mesures

La méthode de correction en ligne constituée d’un filtre adaptatif et d’une interpolation est décrite en VHDL au niveau RTL, simulée sous ADVancedMS et validée par implémentation dans un FPGA de chez Xilinx. Maintenant, le but est de tester la méthode de correction dans un environnement de réel fonctionnement du TIADC : le circuit du TIADC est désormais connecté aux entrées du FPGA. Le convertisseur entrelacé AD8400D, utilisé pour valider la méthode en différé, possède un taux d’échantillonnage à 1 GHz et par conséquent présente des niveaux de sorties en LVDS (Low voltage differential signaling). Les fils de connections entre le TIADC et le FPGA qui devraient supporter du LVDS sont peu disponibles et très coûteux pour notre équipe de recherche. Pour ces raisons, le convertisseur sélectionné afin de valider la méthode en ligne est le AD9218 [83]. Il s’agit d’un convertisseur dual de résolution 10 bits fonctionnant à une fréquence d’échantillonnage maximale égale à 85 MHz. Pour ce convertisseur dual, deux entrées analogiques sont fournies aux deux convertisseurs utilisant la même horloge pour effectuer la conversion. Afin d’assurer l’entrelacement temporel, des modifications sont apportées à la carte d’évaluation de ce circuit pour séparer les deux horloges des deux CAN afin de pouvoir utiliser



(a) Sans correction



(b) Avec correction

FIG. 6.7 – Spectre de sortie du TIADC avant et après calibration utilisant des données expérimentales.

deux horloges complémentées. Un générateur d'impulsions HP8131A est utilisé pour fournir deux sources d'horloge complémentées.

Le nouveau banc de tests utilise les mêmes générateurs de signaux d'entrée et d'horloge, le même analyseur logique que le premier banc de test et ADClab comme outil de caractérisation. Il utilise un FPGA de type Virtex-II PRO (contenant le circuit de la méthode de correction en ligne proposée) un convertisseur de type AD9218 et un générateur d'impulsions pour générer deux horloges complémentées.

Pour des raisons de flexibilité, un multiplexeur est ajouté dans le circuit pour permettre une récupération des données avec ou sans correction. Le schéma du principe est illustré dans la figure 6.8. Les résultats de mesures sans et avec correction sont donnés respectivement dans les

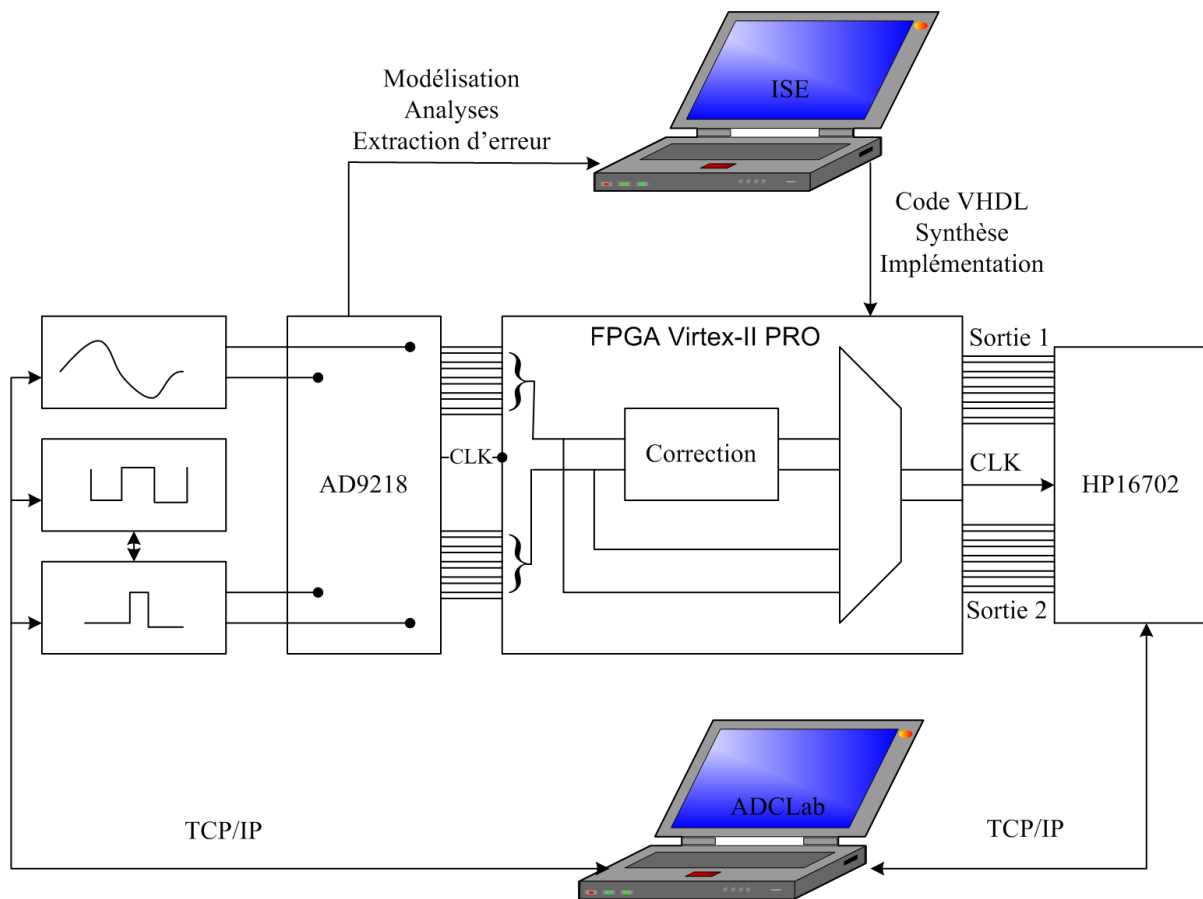


FIG. 6.8 – Banc de mesure pour la méthode en ligne.

figures 6.9 et 6.10. Le spectre de la figure 6.9 montre, par la présence de raies non harmoniques, de fortes disparités entre les deux voies du TIADC. Suite à la correction, le spectre de la figure 6.10 montre une suppression de la raie à la fréquence DC , une diminution de la raies de disparités des gains et des retards d'ouverture mais aussi l'apparition d'un plancher de bruit non uniforme

additionnel au bruit de quantification. Ceci est dû principalement à l'importance des erreurs de

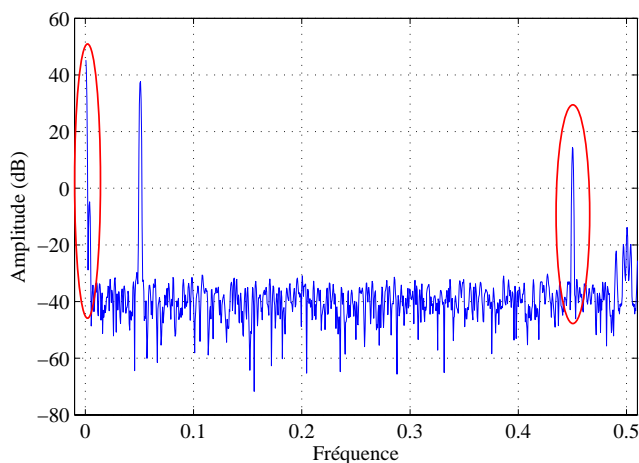


FIG. 6.9 – Spectre de sortie mesurée à la sortie du TIADC avant implémentation de la méthode de correction.

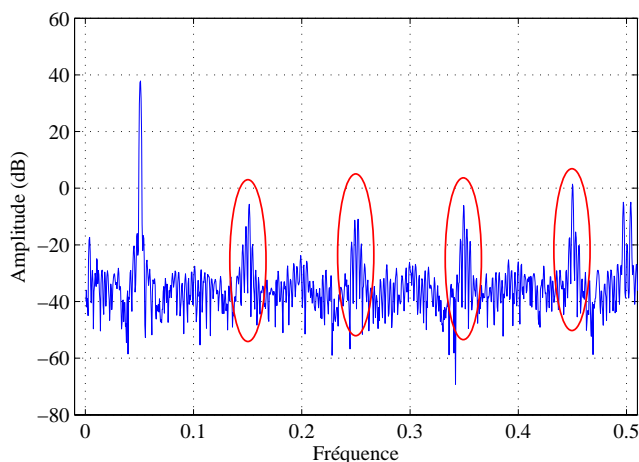


FIG. 6.10 – Spectre de sortie mesurée à la sortie du TIADC après implémentation de la méthode de correction.

retards à l'ouverture au niveau du circuit de conversion de données. En effet, les modifications apportées au circuit pour avoir un fonctionnement entrelacé dans le temps insèrent des retards temporel additionnels. En outre, ces erreurs s'amplifient par les différences au niveau des retards dus au générateurs d'impulsion. Enfin, afin de s'assurer de la bonne interprétation des origines de la non uniformité du plancher du bruit, le modèle du TIADC a été simulé utilisant les mêmes paramètres de la méthode de correction et 10% de différence de retards à l'ouverture entre les

deux voies constituant le TIADC. Les résultats de simulation au niveau du spectre de sortie suite à la correction montrent une non uniformité du niveau de bruit et par la suite, confirment le fait que les erreurs des retards à l'ouverture sont à l'origine du plancher de bruit non uniforme. Ce plancher de bruit non uniforme cause, au niveau des résultats de mesure, une perte de SNR évaluée à 4.6 dB. Cependant, concernant l'évolution du $SFDR$, la figure 6.10 montre une augmentation du $SFDR$ de 13 dB.

6.5 Conclusion

Ce chapitre est l'aboutissement de l'étude des méthodes de correction des erreurs de disparité dans les TIADC. Il est consacré à l'implémentation matérielle des méthodes de correction et à leurs validations fonctionnelles. La première méthode de correction en différé développée sous Matlab est intégrée dans un logiciel de caractérisation des CAN nommé ADCLab. Un premier banc de tests est mis en oeuvre pour extraire des données à la sortie d'un convertisseur entrelacé de résolution 8 bits. Ensuite, les échantillons ont été traités avec ADCLab afin de comparer les sorties sans et avec correction. Suite à cette correction nous avons estimé les caractéristiques suivantes : le nombre de bits effectifs passe de 4.02 bits à 6.36 bits et le $SFDR$ augmente de plus de 23 dB à 50.14 dB pour une fréquence d'échantillonnage de 220 MHz et un signal d'entrée cohérent à 5 MHz.

La méthode en ligne est synthétisée sur plusieurs circuits reconfigurables de différentes familles et générations. Les résultats de synthèse montrent que la consommation des ressources matérielles (slices et LUT) est relativement basse comparée aux nombres de ressources matérielles disponibles dans ces circuits reconfigurables. Après implémentation, nous avons mis l'accent sur l'intérêt du filtre DLMS : sa fréquence maximale de fonctionnement est constante alors que dans le cas du filtre LMS, la fréquence varie en fonction de l'ordre du filtre FIR.

La méthode en ligne fut testée, par des données d'entrées issues des simulations d'un modèle du TIADC, à l'aide d'un processeur intégré. Les résultats d'implémentations et de tests confirment le bon fonctionnement de la méthode.

Enfin, un deuxième banc de tests est mis en place afin de tester la méthode en ligne implémentée dans un FPGA de type Virtex-II Pro. Les problèmes de synchronisation dus au générateur d'impulsions utilisé et aux modifications apportées au circuit du TIADC expliquent la forme des spectres après la correction. Néanmoins, quelques notes positives peuvent être extraites à travers ces validations expérimentales : le $SFDR$ présente une augmentation de 13 dB ce qui valide ces mesures puisque l'objectif que nous avons fixé dès le début consistait à éliminer les raies de disparités. Ensuite, l'amélioration du prototype de test en utilisant un vrai circuit du TIADC est envisagée dans la suite de ces travaux pour améliorer les caractéristiques spectrales

de la sortie corrigée du TIADC et par la suite le *FOM* du convertisseur. Enfin, la méthode de correction en ligne implémentée dans un FPGA peut être améliorée par implémentation sur un ASIC tout en employant un flot de conception ASIC utilisant les mêmes programmes VHDL.

Conclusion générale

Sommaire

7.1	Contexte des travaux de la thèse	153
7.2	Contributions et résultats	154
7.3	Analyses critiques	155
7.4	Perspectives	156

7.1 Contexte des travaux de la thèse

Les travaux menés pendant cette thèse ont porté sur l'étude, la modélisation et la correction des erreurs de disparités des convertisseurs analogique numérique entrelacés dans le temps. Dans un contexte plus général, le but de cette thèse a été l'optimisation du compromis vitesse/résolution des TIADC toute en gardant une puissance consommée assez faible.

Dans un contexte où les systèmes communicants fleurissent, les convertisseurs analogique numérique doivent suivre les demandes des nouvelles tendances des standards de télécommunications. Ces demandes se situent essentiellement au niveau de l'augmentation de la fréquence d'échantillonnage afin de permettre une numérisation au plus proche de l'antenne, réduire le nombre des fréquences intermédiaires et profiter des avantages du traitement numérique. Les différentes architectures classiques des CAN ne peuvent pas répondre à ces exigences. Une solution pour augmenter le taux d'échantillonnage des CAN consiste à entrelacer dans le temps plusieurs convertisseurs. L'amélioration des caractéristiques de ces solutions est l'objet de ces travaux de

thèse.

Dans un premier temps un état de l'art des CAN, de leur fonctionnement, de leur paramètres et de leurs architectures a été présenté. Il a été conclu qu'un seul CAN ne peut pas allier haute résolution et fréquence d'échantillonnage : les convertisseurs parallèles présentent des fréquences d'échantillonnage assez élevées au détriment de la résolution, alors que les convertisseurs $\Sigma\Delta$ ont une résolution importante mais des fréquences d'échantillonnage relativement basses.

7.2 Contributions et résultats

L'état de l'art des convertisseurs conventionnels (de Nyquist et à suréchantillonnage) a stimulé l'étude des structures entrelacées des CAN. Le principe est de faire fonctionner, de manière circulaire, plusieurs convertisseurs pour multiplexer leurs sorties. La fréquence d'échantillonnage est donc multipliée par le nombre de convertisseurs utilisés. Il a été montré que les défauts technologiques des CAN, qui sont à l'origine de raies déterministes pour un seul CAN, s'amplifient dans le cas des TIADC pour donner naissance à des composantes spectrales non harmoniques dont les caractéristiques dépendent du signal d'entrée et du signal d'horloge. L'effet de chaque erreur est analysé séparément afin de mieux cerner les disparité (mismatching) engendrées. Ensuite, la combinaison de ces erreurs montre qu'il faut prendre au sérieux le problème de disparités et faire des efforts tant au niveau modélisation de la structure entrelacée qu'au niveau méthode de calibration de ces convertisseurs.

Pour répondre à ces besoins, un modèle comportemental du TIADC a été réalisé en utilisant le langage VHDL-AMS. L'intérêt de ce langage fut de pouvoir faire interagir des signaux analogiques et numériques dans le même environnement de modélisation et de simulation. Les résultats de simulation confirment le phénomène de disparité entre les canaux d'un TIADC et nous incitent à étudier les méthodes de calibration afin de dégager leurs intérêts.

Ces méthodes de calibration sont classées selon leur mode de fonctionnement (méthode en ligne ou méthode en différé) et selon leur emplacement par rapport au TIADC (méthode analogique ou méthode numérique). Nous nous sommes intéressés à l'étude de quatre méthodes de calibration numériques : deux méthodes en différé basées sur l'utilisation de séquences orthogonales et sur l'estimation des erreurs par FFT, et deux méthodes en ligne utilisant des générateurs de bruit aléatoire. Ces différentes méthodes ont été intégrées dans le modèle du TIADC ou testées à la sortie de ce modèle. Après simulations, des discussions ont été menées pour chaque méthode et un tableau récapitulatif a été présenté afin de les comparer.

Dans le but de répondre à la problématique posée par les TIADC, et face à l'insuffisance des méthodes de correction existantes, nous avons développé deux méthodes de calibration.

La première est une méthode en différé, elle se base sur des calculs trigonométriques de moyennes pour corriger les erreurs d'offset et de gain dans les TIADC. L'erreur d'offset de chaque CAN est fixée à zéro, celle du gain est fixée à la valeur de gain du CAN de référence. La vitesse maximale de fonctionnement n'est pas réduite par la correction qui s'applique théoriquement pour n'importe quel nombre de CAN. L'évolution des paramètres spectraux obtenus aux travers les simulations et les mesures montrent le bon fonctionnement de la méthode proposée.

La deuxième méthode est une méthode adaptative en ligne constituée d'un filtre adaptatif de type LMS qui rapproche les fonctions de transfert des deux CAN et d'une méthode d'interpolation en cosinus qui sert pour la synchronisation. Des améliorations sont ensuite apportées au niveau du filtrage afin de considérer des signaux non stationnaires et d'augmenter le débit maximum du système. Intégrée derrière le modèle du TIADC, les résultats de simulation de cette méthode témoignent des bonnes caractéristiques du système.

Enfin, deux bancs de tests ont été mis en oeuvre pour la validation des méthodes proposées. La méthode en différé a été validée par des données expérimentales utilisant ADCLab comme outil de caractérisation. La méthode en ligne implantée dans un FPGA positionné après un convertisseur entrelacé, a permis l'amélioration du *SFDR* mais des problèmes d'acquisition et de connectique ont fait que le nombre de bits effectifs n'a pas suivi la même évolution.

7.3 Analyses critiques

Les performances des convertisseurs entrelacés dépendent de celles des CAN. Il est indispensable de garder en tête l'évolution des CAN 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 CAN. 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 de CAN utilisés dans le TIADC. Dans le cas de la radio logicielle par exemple, la résolution requise par les CAN est estimée à 18 bits. Les CAN offrant cette résolution ne dépassent pas les quelques MS/s en fréquence d'échantillonnage et le nombre de CAN à entrelacer devient exorbitant pour atteindre la fréquence d'échantillonnage requise. Néanmoins, pour d'autres applications telles que les instruments de mesures et les systèmes Radar, la demande en terme de résolution est moins contraignante et les TIADC présentent une solution prometteuse.

Ainsi, des efforts sont indispensables pour améliorer le facteur de mérite des CAN. Ensuite, les méthodes de correction, telles que celles proposées dans ce manuscrit, doivent se rapprocher au

mieux de ces résolutions.

7.4 Perspectives

Le dernier volet de ce qui a été réalisé durant cette thèse nécessite des améliorations. Les premiers travaux qui s'imposent sont l'amélioration des solutions de correction proposées.

Tout d'abord, la méthode en différé qui sert pour la compensation des erreurs d'offset et de gain peut servir pour des applications en temps réel en la changeant en une méthode en ligne. En effet, la méthode en question se base sur le calcul de moyennes pour l'estimation des erreurs d'offset et de gain de chaque CAN. Ce calcul se fait en différé et pourrait être effectué en ligne par l'application d'algorithmes adaptatifs d'estimation de moyenne.

Ensuite, il y a un intérêt particulier pour l'amélioration de la méthode en ligne pour tenir compte de l'erreur de retard à l'ouverture. La méthode en ligne proposée dans ce manuscrit traite les erreurs de retard à l'ouverture. Mais lorsque ce retard est très important par rapport à la période d'échantillonnage (dépassant les 10%), la méthode proposée est insuffisante.

Il serait bien sûr intéressant de continuer le travail sur la compensation des erreurs de disparité dans les TIADC en améliorant les solutions existantes et proposées, en proposant notamment une correction de l'erreur de retard à l'ouverture. Ces méthodes doivent être intégrées dans un composant programmable de type FPGA, comme nous l'avons fait dans nos travaux, ou de type ASIC pour en améliorer les performances.

Enfin, les travaux présentés dans ce manuscrit se poursuivent dans le cadre d'un projet de Recherche Exploratoire et Innovation REI nommé CAN ENTRELACES qui a débuté en Septembre 2007 avec le Centre Electronique de l'Armement CELAR sous la direction de la Délégation Générale de l'Armement DGA qui a financé mes trois années de thèse. L'objectif de ce projet concerne l'étude et la conception d'un système de conversion réalisant l'échantillonnage d'un signal à une cadence de 6 GHz dont la bande passante analogique maximale est de 5 GHz par la technique d'entrelacement de CAN. Le projet CAN ENTRELACES mettra à la disposition de la DGA un démonstrateur d'acquisition novateur et unique (qui inclut des CAN conçus par la DGA) précis, très large bande et à haute cadence d'échantillonnage. Il rendra possible la réalisation d'intercepteur radar à forte sensibilité et à fort pouvoir séparateur dans une bande instantanée de 2 à 2.5 GHz, quadruplant ainsi la bande passante des intercepteurs actuels de ce type.

Cela ouvrira de nouvelles perspectives de développement de systèmes de réception très large bande à haute cadence d'échantillonnage et dont l'efficacité sera supérieure aux systèmes actuels.

Liste des publications

Revues internationales

1. M. Jridi, G. Ferré, L. Bossuet, B. Le Gal and D. Dallet, Two Time-Interleaved ADC Channel Structure : Analysis and Modeling, *en révision dans IEEE Transactions on Instrumentation and Measurement*.

Conférences internationales avec comité de lecture

1. M. Jridi, D. Dallet, Ch. Rebai, P. Marchegay, Digital Sigma Delta Oscillator : Design Consideration, In *Design of Circuits and Integrated Systems*, Décembre 2004.
2. M. Jridi, R. Shirakawa and D. Dallet, Aperture jitter and timing skew analyses in ADC structures, In *4th Imeko Symposium on new technologies in measurement and instrumentation and 10th Workshop on ADC modelling and testing*, Septembre 2005.
3. M. Jridi, G. Monnerie, L. Bossuet, D. Dallet, Two time-interleaved ADC channel structure : analysis and modeling, In *IEEE Instrumentation and Measurement Technology Conference*, Avril 2006.
4. M. Jridi, L. Bossuet, B. Le Gal and D. Dallet, An offset and gain calibration method for time-interleaved analog to digital converters, In *IEEE International Conference on Electronics, Circuits and Systems*, Décembre 2006.
5. M. Jridi, L. Bossuet, B. Le Gal and D. Dallet, New Adaptive Calibration Method for Time Interleaved Analog to Digital Converters, In *In IEEE NEWCAS*, Août 2007.
6. G. Ferré, M. Jridi, L. Bossuet, B. Le Gal and D. Dallet, A New Orthogonal Online Digital Calibration for Time-Interleaved Analog-to-Digital Converters, In *IEEE International Symposium on Circuits and Systems*, Mai 2008.

Conférences nationales avec comité de lecture

1. M. Jridi, L. Bossuet, B. Le Gal and D. Dallet, Les convertisseurs analogique numérique entrelacés dans le temps : théorie et nouvelles tendances des méthodes de correction d'erreurs, In *Journées Nationales du Réseau Doctoral en Microélectronique*, Mai 2007.

Rapports de recherche

1. M. Jridi, Rapport technique - Projet IXL/CELAR - , Implémentation des cartes d'acquisition dans un analyseur logique, Laboratoire IMS - Département COFI - Groupe Conception, Octobre 2005.

2. D. Dallet and M. Jridi, Rapport technique - Projet IXL/CELAR - , Développement d'une version CanTesT sous Matlab, étude du banc de filtre et des structures entrelacés, Laboratoire IMS - Département COFI - Groupe Conception, Juin 2006.
3. M. Jridi, Rapport technique - Projet IXL/CELAR -, Amélioration de la définition des paramètres spectraux et traitement de raies de bruit en bande canal, Laboratoire IMS - Département COFI - Groupe Conception, Octobre 2006.
4. D. Dallet and M. Jridi, Rapport technique - Projet REI - DGA, Étude de techniques d'entrelacement pour système d'acquisition numérique rapide et ultra large bande, Laboratoire IMS - Département COFI - Groupe Conception, Août 2007.

ANNEXE A

Etude comparative de la co-simulation et de la simulation mixte

Dans le dernier chapitre, nous avons mis l'accent sur la nécessité d'une description mixte dès lors qu'il s'agit de modéliser des systèmes combinant des signaux analogiques et numériques. Les TIADC font partie de ces systèmes, ils seront modélisés de la même sorte. La première finalité est de pouvoir simuler les signaux analogiques, discrets et numériques du système dans un même environnement pour vérifier son bon fonctionnement. La deuxième finalité consiste à fournir une implémentation matérielle de la partie post-correction numérique. Ces deux points peuvent être obtenus par une co-simulation ou une simulation mixte. Nous détaillons dans la suite le principe de chacune de ces deux méthodologies tout en insistant sur leurs différences.

1 : La co-simulation

1.1 : Principe

Le terme de co-simulation, matérielle-logicielle, désigne cette simulation conjointe. Nous distinguons deux types de co-simulations, selon le niveau auquel elle s'effectue. La co-simulation de bas niveau qui fait intervenir de l'assembleur et du VHDL et la co-simulation de haut niveau qui utilise du VHDL avec un code écrit en C ou des entrées schématiques faites sous Matlab et transformées ensuite en description matérielle. Nous nous intéressons à cette dernière forme de co-simulation. Matlab fournit des descriptions au niveau fonctionnel, structurel ou comportemental grâce à sa boîte à outil schématique System. L'intérêt majeur réside dans sa simplicité d'utilisation, sa flexibilité ainsi que dans la possibilité de faire des descriptions schématiques en flottant ou en virgule fixe. Simulink fournit des modèles élémentaires paramétrables (registres, additionneurs, multiplieurs,...) pour réaliser des composants de modèle. Les différents composants sont liés schématiquement pour aboutir au modèle global du système. Les sorties globales et intermédiaires du modèle peuvent être stockées dans des mémoires pour s'assurer du bon fonctionnement du modèle. Une fois le fonctionnement validé, des boîtes à outils de Xilinx, System Generator, similaires à celles de Simulink peuvent être utilisées pour passer d'une modélisation comportementale ou structurelle à une description matérielle. Dans la figure 7.1, un exemple d'une description matérielle d'un filtre FIR sous System Generator est illustré. L'outil permet aussi de choisir la cible matérielle d'implémentation, le logiciel de simulation et de synthèse. Une estimation de ressources matérielles consommées est fournie afin d'évaluer la qualité du modèle. L'outil renvoie en sortie un fichier VHDL synthétisable être ensuite implémenté dans des circuits programmables. Un exemple d'utilisation de cette méthodologie est donné dans [18] et [84].

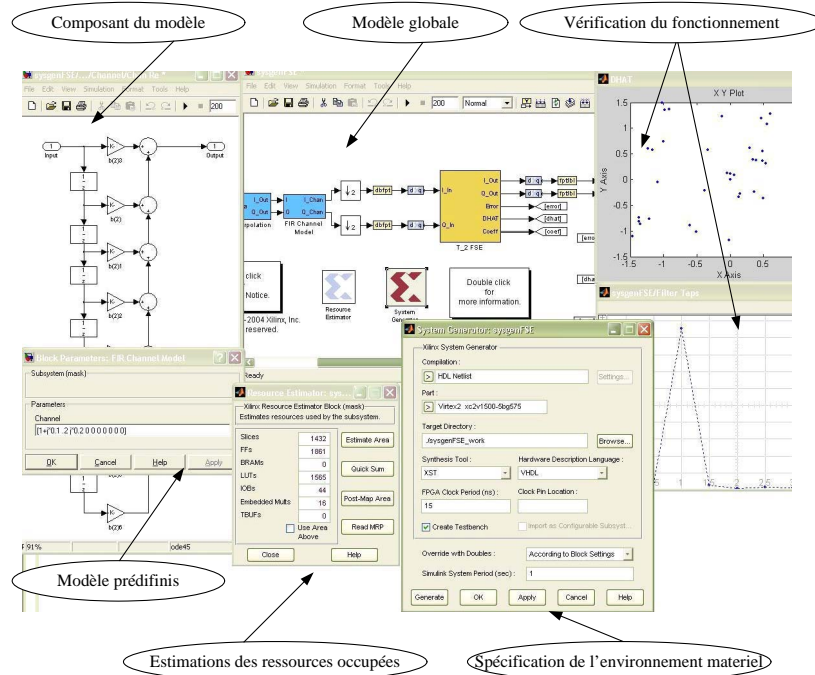


FIG. 7.1 – Environnement de la co-simulation sous System Generator.

1.2 : Limites de la co-simulation

Malgré la flexibilité et la simplicité, la co-simulation présente plusieurs limites surtout quand il s'agit de l'appliquer pour des systèmes à comportement mixte. En effet, pour faire réagir les signaux analogiques et numériques dans le même environnement, nous utilisons des signaux discrétisés à pas très faible par rapport à la période du signal d'entrée. Le signal reste discret, mais la différence entre un état et l'état suivant est infiniment petite.

Pour appliquer cette méthode à la modélisation d'un TIADC, nous proposons d'abord une modélisation de l'échantillonneur bloqueur. Ce système d'échantillonnage a été modélisé sous Simulink pour montrer l'effet des erreurs de jitter et de retard d'ouverture, [35]. La figure suivante 7.2 illustre le système d'échantillonnage proposé qui est composé de trois sources d'entrée : un générateur du signal sinusoïdale d'entrée dont l'amplitude, la fréquence et la phase sont paramétrables, un générateur de signal carré d'horloge et un générateur de bruit aléatoire modélisant le jitter. Ce dernier générateur a une moyenne δt décrivant l'erreur de retard d'ouverture. Pour modéliser les erreurs temporelles, le signal d'horloge n'est plus un signal carré théorique mais désormais un signal carré concaténé à un bruit aléatoire. En effet, grâce au bloc *Variable Transport Delay*, la période du signal d'horloge n'est plus régulière. Des retards fixes et aléatoires viennent s'ajouter à ce signal. Le signal obtenue est considéré comme *trigger* de l'échantillonneur bloqueur S/H.

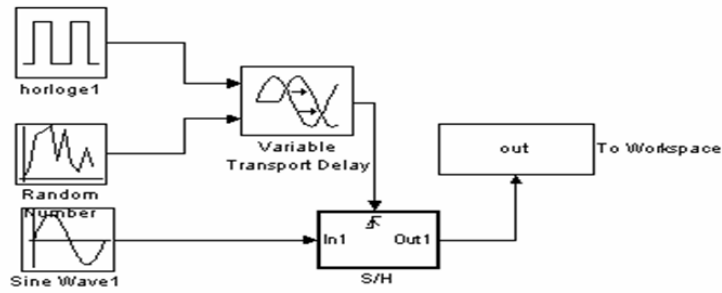


FIG. 7.2 – Système d'échantillonnage.

Pour mettre en évidence l'effet jitter, nous échantillons le signal d'entrée à $f_s = f_0$. En absence de jitter, nous récupérons en sortie le même point. En présence du jitter, l'échantillonnage ne se fait plus au même point : c'est la valeur aléatoire du jitter qui fixe le point échantillonné à chaque période. La sortie du modèle est donc proportionnelle au jitter.

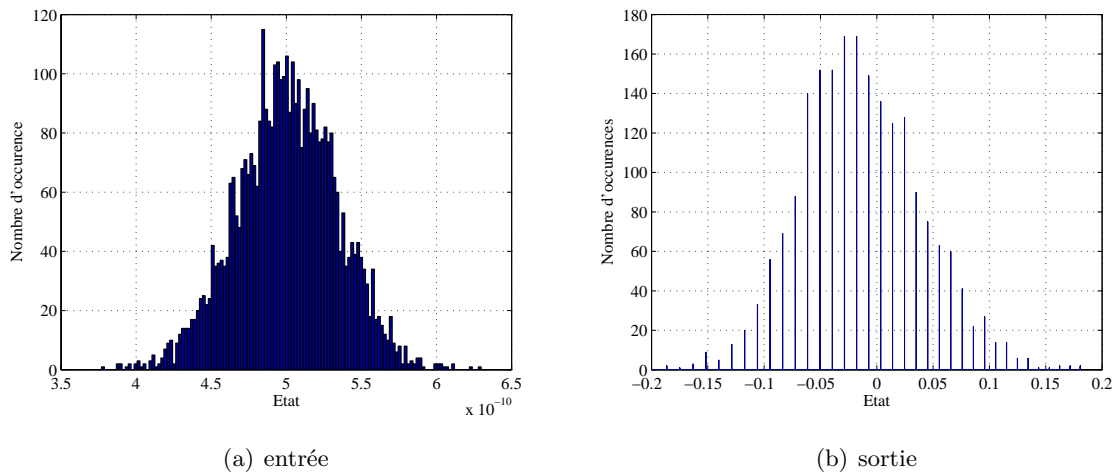


FIG. 7.3 – Entrée/sortie du système d'échantillonnage.

La figure 7.3 montre les histogrammes du générateur du bruit aléatoire (l'entrée) et de la sortie du modèle. Les deux histogrammes devraient être similaires. Les différences sont dus au fait que nous utilisons un signal d'entrée discret et par la suite le nombre d'états est fini. De plus, nous utilisons une fréquence d'échantillonnage pour tous le système et une fréquence de calcul pour avoir plus de précision. Ceci explique la diminution de nombre d'états en sortie. Dans un TIADC cette différence augmente les disparités entre les différents convertisseurs. Mis à part la précision, cette méthodologie présente d'autres inconvénients. L'utilisation obliga-

toire des modèles prédéfinis de Simulink et de System Generator, diminue les degrés de liberté des concepteurs. Généralement, ces modèles prédéfini ne présentent pas tous les paramètres et les options nécessaire et l'outil ne donne pas la possibilité d'en ajouter. Enfin, le fichier VHDL fourni par cette méthodologie n'est pas de haut niveau dans le sens où il ne peut pas être interprété. En outre des bibliothèques propres à Xilinx (et propres à Altera pour DSPBuilder) sont employées, ce qui permet de tester le système sur des cibles programmables de Xilinx (d'Altera) mais ceci n'est pas adapté pour toutes les autres familles de FPGA ni pour les ASIC (Application Specific Integrated Circuit).

2 : La simulation mixte

2.1 : Principe

La simulation mixte fait interagir des signaux discret en temps et en amplitude avec des signaux continus. Ici se pose le problème de la synchronisation des deux noyaux du simulateur. En effet, il faut que les noyaux numériques et analogiques puissent se synchroniser. Dans la plupart des applications chaque noyau gère sa propre horloge mais doit conserver des liens avec l'autre pour assurer la cohérence de l'évaluation. Pour l'action du noyau numérique sur le noyau analogique, un événement sur un SIGNAL doit pouvoir provoquer une évaluation analogique. Comme le pas temporel de la simulation analogique est variable il n'y a aucune assurance qu'au moment de l'événement numérique il y ait un point analogique. Il faut donc forcer le calcul d'un point de simulation analogique au moment d'un événement. L'instruction BREAK est la solution prévue par VHDL-AMS.

Par ailleurs, la simulation mixte s'installe dans le cadre de la conception descendante (TOP-DOWN). La conception descendante est le fait d'effectuer une conception en commençant par un haut niveau d'abstraction (description fonctionnelle) pour passer par étapes successives à une réalisation physique par interconnexion des composants de base . Cette approche permet de placer le choix de la technologie le plus tard possible dans le cycle de conception. De la même façon une modification de technologie ne remet pas en cause les premières étapes de la conception. Elle s'oppose à l'approche ascendante (BOTTOM-UP) qui commence par concevoir technologiquement toutes les briques de base pour ensuite les assembler.

L'outil Matlab sera d'abord utilisé pour la modélisation fonctionnelle : développement de fonctions prédéfinies pour vérifier le bon fonctionnement. Ensuite des algorithmes appropriés sont créés pour s'assurer de la bonne compréhension des fonctions prédéfinies . De ces algorithmes découlent des descriptions structurelles à l'aide de Simulink.

Après vérifications, le système est ensuite modélisé avec VHDL-AMS sous ADVanceMS dans

un environnement de simulation mixte. La partie numérique est ensuite extraite pour l'implémenter sur un FPGA à l'aide de l'outil ISE de Xilinx. *2.2 : Avantages de la simulation mixte* Pour vérifier l'intérêt de cette méthodologie par rapport à celle utilisant la co-simulation, nous proposons de modéliser le même système d'échantillonnage décrit dans la section précédente. La génération d'une horloge avec un bruit de jitter sera montré dans l'annexe C. L'échantillonneur bloqueur est modélisé comme le montre l'algorithme suivant :

```
-----
LIBRARY DISCIPLINES;
USE DISCIPLINES.electromagnetic_system.all;
-----
```

```
ENTITY sampler IS
  PORT ( SIGNAL clock : IN BIT;
        SIGNAL Nout : OUT REAL := 0.0;
        TERMINAL Pimp, Pinn: ELECTRICAL);
END ENTITY sampler;
```

```
-----
ARCHITECTURE behaviour OF sampler IS
  QUANTITY Vin ACROSS Pimp TO Pinn;
  BEGIN
  BREAK ON clock;
  sampling : PROCESS (clock)
  BEGIN
    IF (clock = '0') THEN
      Nout <= Vin; END IF;
  END PROCESS sampling;
END ARCHITECTURE behaviour;
```

La figure 7.4 montre les histogrammes du bruit à l'entrée et à la sortie du système d'échantillonnage. Contrairement aux histogrammes obtenus en co-simulation, nous remarquons que le nombre d'états à l'entrée et à la sortie est le même. Nous remarquons qu'il y a plus de précision et que la différence entrée/sortie est pratiquement nulle.

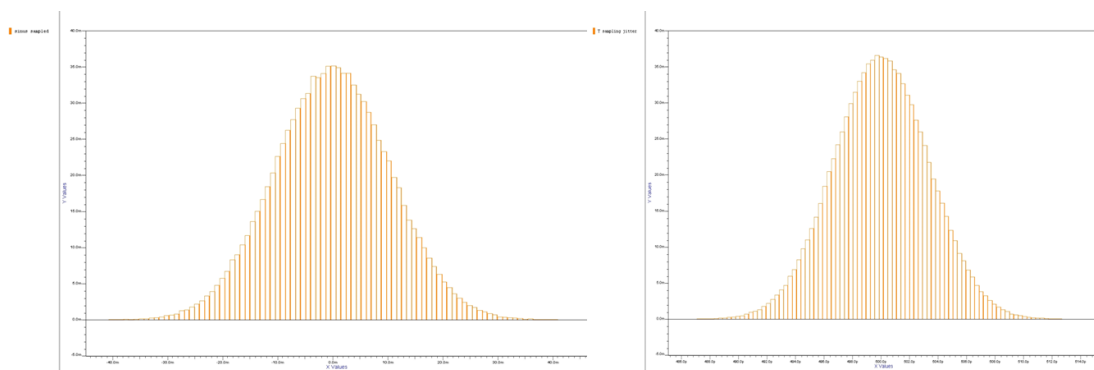


FIG. 7.4 – Entrée/sortie du système d'échantillonnage modélisé en VHDL-AMS.

De plus, la précision de la simulation mixte présente des avantages d'ordre économique et technique. En effet, la simulation mixte utilisant VHDL-AMS comme langage de description profite

de la standardisation de ce dernier : le fait de disposer d'un langage commun à tous les acteurs permet le développement de méthode de conception par réutilisation (re-use) de bloc appelés IP (Intellectual Property). Les investissements sont plus sûrs du fait d'un standard industriel, de l'indépendance par rapport au fondateur et de l'indépendance par rapport à un fournisseur d'outils de conception assistée par ordinateur [85].

ANNEXE B

Modélisation mixte du TIADC

Les sections précédentes ont permis la définition de l’environnement de description mixte. Le choix est fixé sur l’utilisation du langage VHDL-AMS comme langage et d’ADVanceMS comme simulateur mixte. Dans cette section nous décrivons le modèle du TIADC conformément à la figure 7.5. Les trois blocs essentiels dans cette description sont les générateurs de signaux, le convertisseur proprement dit et le multiplexeur.

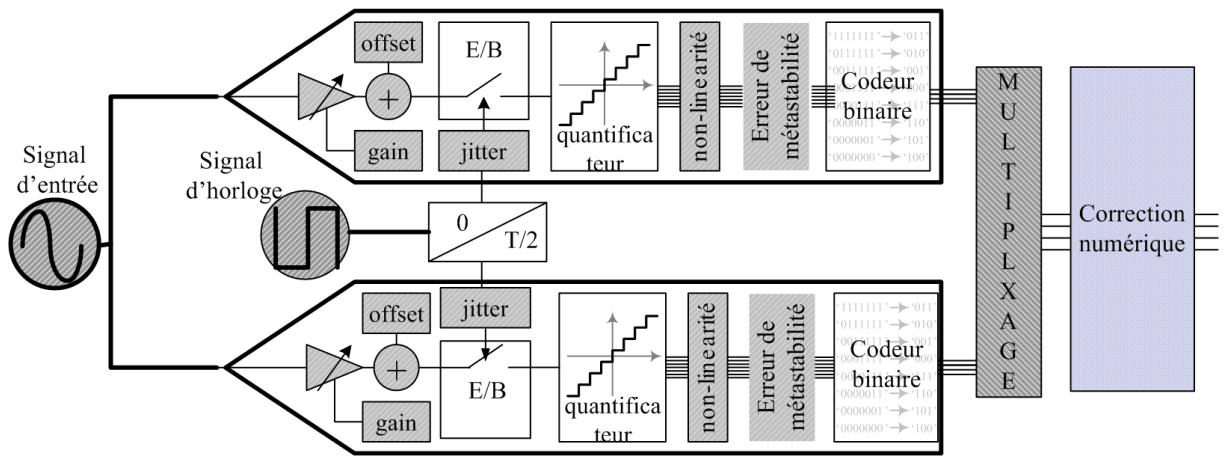


FIG. 7.5 – Modèle du TIADC.

1 : Génération des signaux Les deux signaux en question sont le signal d’entrée appliqué au CAN et le signal d’horloge qui génère l’échantillonnage. Nous décrivons dans les paragraphes suivants leur modélisation en VHDL-AMS.

1.1 : Génération du signal d’horloge

En théorie, le signal d’horloge est un signal carré périodique. Ce signal va commander l’échantillonneur/bloqueur. Les imperfections sur l’instant d’échantillonnage dues au jitter, sont définies comme des caractéristiques propres au générateur d’horloge. Ce bruit de phase de l’oscillateur est un bruit gaussien qui se superpose au signal théorique de l’horloge. Le langage VHDL-AMS ne fournit pas ce type de distribution. Il a donc fallu utiliser des fonctions existantes pour créer une distribution gaussienne. Dans [?], une distribution gaussienne est exprimée en fonction de deux distributions uniformes ($u1, u2$) entre $]0, 1]$ conformément à l’équation 7.1

$$v = m + \sigma \cdot \sqrt{-2 \cdot \log(u1)} \cdot \sin(2 \cdot \pi \cdot u2) \quad (7.1)$$

v est donc une distribution gaussienne de moyenne m et d'écart type σ . Ce principe a été repris dans [86] pour générer une distribution gaussienne sur VHDL-AMS (fonction *gauss*) décrite dans le programme suivant.

```

-----
-----
FUNCTION gauss (std_deviation: REAL; seed : REAL_VECTOR(0 TO 4))
RETURN REAL_VECTOR IS
VARIABLE rand_1 : REAL :=1.0;
VARIABLE rand_2 : REAL :=1.0;
VARIABLE seed_1 : POSITIVE := INTEGER(seed(1));
VARIABLE seed_2 : POSITIVE := INTEGER(seed(2));
VARIABLE seed_3 : POSITIVE := INTEGER(seed(3));
VARIABLE seed_4 : POSITIVE := INTEGER(seed(4));
VARIABLE rand : REAL_VECTOR (0 TO 4);
BEGIN
UNIFORM(seed_1,seed_2,rand_1);
UNIFORM(seed_3,seed_4,rand_2);
rand(0) := std_deviation*SQRT(-2.0*LOG(rand_1))*SIN(MATH_2_PI*rand_2);
rand(1) := REAL(seed_1);
rand(2) := REAL(seed_2);
rand(3) := REAL(seed_3);
rand(4) := REAL(seed_4);
RETURN (rand);
END FUNCTION;
-----
-----

```

L'utilisateur fixe la valeur de l'écart type et un vecteur à 5 composantes (contenant les états précédents du générateur de nombres aléatoires). Le vecteur seed(1 TO 4) est un vecteur de variables caractéristiques des séquences aléatoires et sert à leur initialisation. Nous utilisons la fonction UNIFORM de la librairie *math – real* avec $m = 0$. La fonction retourne un nombre gaussien rand(0) et renvoie les états des générateurs de nombres pseudo aléatoires : rand(1 TO 4). Un exemple de distribution gaussienne générée par cette fonction utilisant un écart type égale à 1 est montré dans la figure 7.6.

Cette distribution gaussienne se superpose avec un signal d'horloge théorique pour générer un signal d'horloge contenant un jitter. En spécifiant une période, un rapport cyclique et des conditions d'initialisation nous pouvons ainsi générer un signal d'horloge conformément au programme suivant.

```

-----
-----
LIBRARY TIADC;
USE TIADC.gauss.all;
-----
-----
ENTITY Latch_gene IS
GENERIC(cyclic_ratio : REAL := 0.5;
delta : REAL := 0.0;
sigma : REAL := 0.0;
init : REAL_VECTOR (0 TO 4) := (0.0, 2147483562.0,214.0,546.0,110785.0));
PORT ( SIGNAL period : IN REAL := 0.0;
SIGNAL latch_out : OUT BIT := '0');
END ENTITY Latch_gene;
-----
-----
ARCHITECTURE behaviour OF Latch_gene IS

```

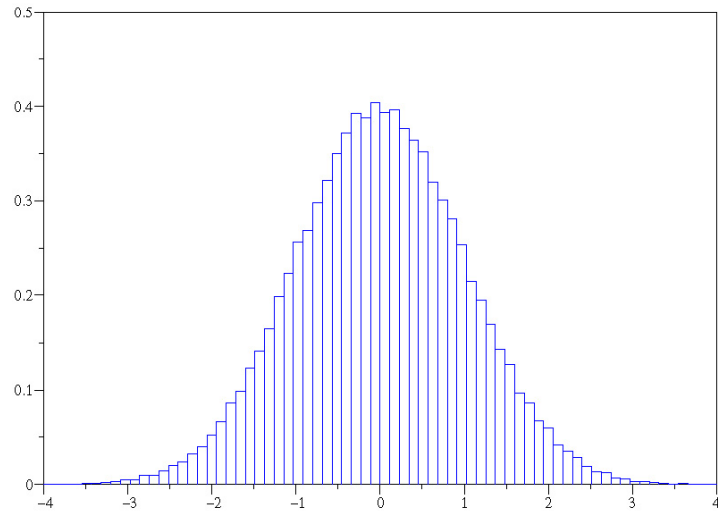



FIG. 7.6 – Histogramme de la distribution gaussienne.

```

BEGIN
  latch_generation : PROCESS
    VARIABLE rand : REAL_VECTOR (0 TO 4) := init;
    VARIABLE test : BIT := '1';
    BEGIN
      IF period > 0.0 THEN
        IF test = '1' THEN test := '0'; WAIT FOR period*delta; END IF;
        IF sigma /= 0.0 THEN rand := gauss(sigma,rand); END IF;
        IF ((cyclic_ratio + rand(0)/100.0)*period > 0.0) AND (((1.0-cyclic_ratio) - rand(0)/100.0)*period > 0.0) THEN
          latch_out <= '1'; WAIT FOR (cyclic_ratio + rand(0)/100.0)*period;
          latch_out <= '0'; WAIT FOR ((1.0-cyclic_ratio) - rand(0)/100.0)*period;
        ELSE
          latch_out <= '1'; WAIT FOR cyclic_ratio*period;
          latch_out <= '0'; WAIT FOR (1.0-cyclic_ratio)*period;
          ASSERT ((cyclic_ratio + rand(0)/100.0)*period > 0.0)
            REPORT "NEGATIV PULSE WIDTH UP" SEVERITY warning;
          ASSERT (((1.0-cyclic_ratio) + rand(0)/100.0)*period > 0.0)
            REPORT "NEGATIV PULSE WIDTH DOWN" SEVERITY warning;
        END IF;
      ELSE WAIT UNTIL period > 0.0;
    END IF;
  END PROCESS latch_generation;
END ARCHITECTURE behaviour;
-----
-----

```

Une fois cette horloge générée, nous pouvons générer une deuxième horloge avec un jitter différent et retardée d'une demi période pour l'entrelacement de deux convertisseurs. Cet appel se fait par instanciation du composant latch-gene. Le paramètre générique delta permet de retarder la deuxième horloge d'une demi période. Ce modèle peut facilement être modifié pour générer plusieurs signaux d'horloge et ainsi entrelacer plusieurs convertisseurs.

```

-----
-----
--  _|-----|-----|-----|-----|

```

```

-- Latch_out(0)
-- -----|-----|-----|-----
-- Latch_out(1)
-----
latch_0_gen : ENTITY latch_gene(behaviour)
GENERIC MAP ( cyclic_ratio => cyclic_ratio, sigma => sigma(0), delta => skew(0)/100.0,
init => (0.0, 2147483562.0,214.0,546.0,110785.0))
PORT MAP ( period => period, latch_out => latch_out(0));

latch_1_gen : ENTITY latch_gene(behaviour)
GENERIC MAP ( cyclic_ratio => cyclic_ratio, sigma => sigma(1), delta => (1.0/2.0 +
skew(1) /100.0), init => (0.0, 214762.0,107214.0,518546.0,118585.0))
PORT MAP ( period => period, latch_out => latch_out(1));
END ARCHITECTURE behaviour_2;
-----
-----

```

1.2 : *Génération du signal d'entrée* Dans cette partie, il s'agit de générer un signal de forme sinusoïdale dont l'équation est :

$$s(t) = A \times \sin(\omega t + \varphi) \tag{7.2}$$

La bibliothèque *math – real* de VHDL-AMS fournit la fonction sinus ainsi que la constante *math – 2 – pi* pour désigner 2π . Pour le paramètre *t* désignant le temps, l'équivalent en VHDL-AMS est la fonction *NOW*. Ainsi, générer le signal d'entrée revient à générer les séquences *X* selon le code suivant :

```

-----
X=amplitude*sin(math-2*pi*fréquence*NOW+phi)
-----

```

Dans [86], une étude complète sur les différentes sources de bruits qui peuvent agir sur le signal d'entrée est illustrée. Ces bruits sont le bruit en amplitude sous la forme d'un bruit blanc ou d'un bruit en $1/f$ et le bruit de phase sous la forme d'un bruit en $1/f^2$ ou en $1/f^3$.

2 : Le convertisseur Le convertisseur, comme nous l'avons décrit dans le chapitre précédent, est constitué d'un échantillonneur bloqueur, d'un quantificateur et d'un codeur binaire. L'échantillonneur est illustré dans la partie précédente, le quantificateur et le codeur sont détaillés dans cette section.

2.1 : *L'échantillonneur bloqueur* Le principe de l'échantillonnage a été détaillé dans le premier chapitre. Il s'agit de relever à instants précis des échantillons du signal d'entrée. Les instants d'échantillonnage sont fixés par les fronts montants et descendants de l'horloge bruitée. Le signal à échantillonner présente une erreur d'offset et de gain. Pour un TIADC le processus d'échantillonnage se fait comme suit :

```

-----
-----

```

```

sampling_with_error : PROCESS (latch_int)
BEGIN
  IF ((latch_int(0)'EVENT) AND (latch_int(0) = '0')) THEN
N_smp(0) <= Gain(0)*(amplitude*sin(MATH_2_PI*frequency*NOW)+N_corr_I)+Offset(0); END IF;
  IF ((latch_int(1)'EVENT) AND (latch_int(1) = '0')) THEN
N_smp(1) <= Gain(1)*(amplitude*sin(MATH_2_PI*frequency*NOW)+N_corr_Q)+Offset(1); END IF;
END PROCESS sampling_with_error;
-----
-----

```

La liste de sensibilité du process en question est défini par l'horloge utilisée pour l'échantillonnage. Les instructions au sein du process ne sont traitées que dans le cas d'un évènement sur les signaux d'horloges.

2.2 : Le quantificateur L'opération de quantification consiste en l'approximation de chaque valeur du signal échantillonné par un multiple entier du pas de quantification q . L'association entre les valeurs réelles du signal d'entrée et le code thermomètre de sortie du quantificateur se fait conformément au programme suivant.

```

-----
-----
LIBRARY IEEE;
USE IEEE.math_real.all;
LIBRARY DISCIPLINES;
USE DISCIPLINES.electromagnetic_system.all;
-----
ENTITY N_bits_Quantifier IS
  GENERIC (
    step      : REAL      := 1.0;
    N_bits    : INTEGER   := 3);
  PORT (
    SIGNAL    Nin          : IN REAL;
    SIGNAL    thermo_code  : OUT BIT_VECTOR((2**N_bits-2) DOWNT0) := (OTHERS =>'0'));
END ENTITY N_bits_Quantifier;
-----
ARCHITECTURE behaviour_bipolar_midtread OF N_bits_Quantifier IS
CONSTANT offset : REAL := -((2.0**(N_bits-1))-0.5)*step;
BEGIN
  codage : PROCESS (Nin)
  BEGIN
    thermo_code <= (OTHERS =>'0');
    FOR i IN 0 TO ((2**N_bits)-1) LOOP
      IF i /= 0 THEN thermo_code(i-1) <= '1'; END IF;
      EXIT WHEN (Nin < (offset + REAL(i)*step));
    END LOOP;
  END PROCESS codage;
END ARCHITECTURE behaviour_bipolar_midtread;
-----
-----

```

2.3 : Le codeur binaire Le codeur binaire a pour but de transformer le code thermomètre de sortie du quantificateur en un code binaire. La longueur du code fixe la résolution théorique du convertisseur.

```

-----
-----
LIBRARY IEEE;
USE IEEE.math_real.all; USE IEEE.std_logic_1164.all;

```

```

USE IEEE.std_logic_signed.all; USE IEEE.std_logic_arith.all;
-----
ENTITY N_bits_Binary_Coder IS
    GENERIC (
        N_bits      : INTEGER      := 4);
    PORT (
        SIGNAL      thermo_code    : IN BIT_VECTOR((2**N_bits-2) DOWNT0 0);
        SIGNAL      binary_code    : OUT STD_LOGIC_VECTOR(N_bits-1 DOWNT0 0) := (OTHERS =>'0'));
END ENTITY N_bits_Binary_Coder;
-----
ARCHITECTURE behaviour_signed OF N_bits_Binary_Coder IS
BEGIN
    coder : PROCESS (thermo_code)
        VARIABLE S          : BIT := '0';
        VARIABLE counter    : REAL := 2.0**(N_bits-1)-1.0;
        VARIABLE i          : INTEGER := 2**N_bits-2;
        VARIABLE stop       : BIT := '1';
    BEGIN
        IF S = '0' THEN      REPORT LF&"SIGNED BINARY CODING" SEVERITY NOTE; S:='1'; END IF;
        FOR i IN 2**N_bits-2 DOWNT0 0 LOOP
            IF ((thermo_code(i) = '0') AND (stop = '1')) THEN counter := counter - 1.0; END IF;
            IF ((thermo_code(i) = '1') AND (stop = '1')) THEN stop := '0'; END IF;
        END LOOP;
        binary_code <= conv_std_logic_vector(integer(counter), N_bits);
        counter := 2.0**(N_bits-1)-1.0;    stop := '1';
    END PROCESS coder;
END ARCHITECTURE behaviour_signed;
-----

```

Un exemple de simulation du signal échantillonné, du code thermomètre et du code binaire est illustré dans la figure 7.7.

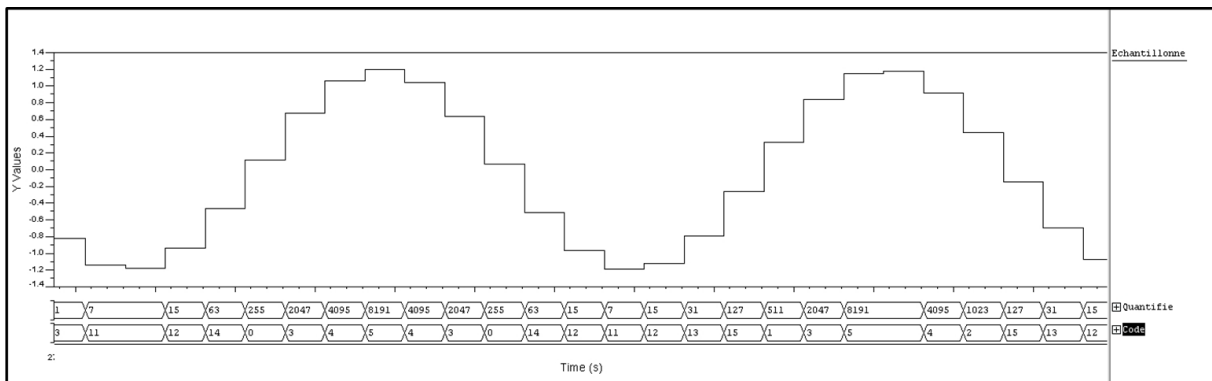


FIG. 7.7 – Sorties du quantificateur et du codeur binaire.

3 : Le multiplexeur Le multiplexeur sert d'aiguillage de l'information selon une entrée supplémentaire appelée entrée de sélection. Dans notre cas, le multiplexeur sert à grouper les données sous la contrainte du front de l'horloge. Sur les fronts montants, la sortie du TIADC est associée à celle du premier CAN. Sur les fronts descendants, la sortie du TIADC est associée à celle du deuxième CAN. Le modèle du multiplexeur est présenté ci-dessous.

```

-----
Modèle du multiplexeur
-----
PROCESS (latch_int)

```

```
BEGIN
  IF ((latch_int(0)'EVENT) AND (latch_int(0)='1')) THEN binary_code <=binary_code_0 ;
  ELSIF ((latch_int(1)'EVENT) AND (latch_int(1) ='1')) THEN binary_code <=binary_code_1_décalé_1; END IF;
END PROCESS;
-----
-----
```

ANNEXE C

Description des méthodes de correction

1 : Méthode de correction par choix aléatoire des canaux

Ici, nous montrons le générateur de nombres pseudo aléatoires.

```
-----  
-----Déclaration des bibliothèques-----  
-----  
Library IEEE;  
use IEEE.math_real.all;  
use IEEE.std_logic_1164.all;  
-----  
-----Déclaration de l'entité-----  
-----  
entity random is  
port ( clock : in std_logic;  
       selection : out std_logic_vector(3 downto 0));  
end random;  
-----  
-----Déclaration de l'architecture-----  
-----  
architecture behav of random is  
begin  
  process (clock)  
variable unif : real;  
variable seed1:integer:=345688; -- initialisation  
variable seed2:integer:=7654354; -- initialisation  
begin  
Uniform(seed1,seed2,unf); -- Sortie  
if ( unif <= 25.0e-2) then -- Quantification  
  selection <= "0001";  
elsif ( unif <= 50.0e-2) then  
  selection <= "0010";  
elsif ( unif <= 75.0e-2) then  
  selection <= "0011";  
else selection <= "0100";  
end if;  
  end process;  
end behav;  
-----  
-----Fin du programme-----  
-----
```

La bibliothèque *IEEE.math – real.all* contient la fonction *Uniform(seed1, seed2, unf)* qui génère des nombres pseudo aléatoires uniformément répartis dans l'intervalle [0, 1].

L'histogramme de cette sortie montre qu'il s'agit bien de nombres aléatoires uniformément répartis. Les éléments de sortie du générateur développé sont des réels. Une quantification est donc

nécessaire pour utiliser ce générateur dans la sélection des convertisseurs dans un TIADC. La sortie quantifiée du générateur contient donc un ensemble fini de valeurs, $\{1, 2, 3, 4\}$. Chacune de ces valeurs est associée à un convertisseur.

2 : Méthode de correction purement numérique Ici nous montrons les algorithmes de correction d'offset et de gain de méthode en ligne étudiée dans le quatrième chapitre et publié dans [40].

```

-----
-----Description de la correction d'offset-----
-----
LIBRARY IEEE;
USE IEEE.math_real.all;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_signed.all;
USE IEEE.std_logic_arith.all;
-----

ENTITY offset_correction IS
    GENERIC(N_bits      : INTEGER := 8;
           N_corr       : INTEGER := 4);
    PORT ( SIGNAL      clock      : IN BIT;
          SIGNAL      NRST       : IN BIT;
          SIGNAL      EN         : IN BIT;
          SIGNAL      binary_in_0 : IN STD_LOGIC_VECTOR(N_bits-1 DOWNT0 0);
          SIGNAL      binary_in_1 : IN STD_LOGIC_VECTOR(N_bits-1 DOWNT0 0);
          SIGNAL      binary_out_0 : INOUT STD_LOGIC_VECTOR(N_bits-1 DOWNT0 0);
          SIGNAL      binary_out_1 : INOUT STD_LOGIC_VECTOR(N_bits-1 DOWNT0 0));
END ENTITY offset_correction;
-----

ARCHITECTURE behaviour_2_channel OF offset_correction IS
BEGIN
    PROCESS (EN)
    BEGIN
        ASSERT (EN = '0')
        REPORT "OFFSET CORRECTION ACTIVE" &LF
        SEVERITY WARNING;
        ASSERT (EN = '1')
        REPORT "OFFSET CORRECTION INACTIVE" &LF
        SEVERITY WARNING;
    END PROCESS;

    PROCESS(clock, NRST)
-----
-----Déclaration des variables intermédiaires-----
-----
BEGIN
    IF NRST='0' THEN binary_out_0 <= (OTHERS => '0'); binary_out_1 <= (OTHERS => '0');
    ELSE
        IF(clock'EVENT AND clock='1') THEN
            IF EN = '0' THEN

```

```

-- SOUSTRACTION (N_bits -> N_bits)
int_2 := binary_in_0 - int_1;
-- MULTIPLICATION CONSTANT (N_bits -> N_corr+N_bits)
IF int_2(N_bits-1) = '1' THEN  --- IF int_2(N_bits+1) = '1' THEN
  int_2 := int_2-ONE;
  int_2:= NOT int_2;
  FOR i IN 0 TO N_bits-1 LOOP int_3(i) := int_2(i); END LOOP;
  FOR i IN N_bits TO N_corr+N_bits-1 LOOP int_3(i) := '0'; END LOOP;
  int_3 := NOT int_3;
  int_3 := int_3+ONEb;
ELSE
  FOR i IN 0 TO N_bits-1 LOOP int_3(i) := int_2(i); END LOOP;
  FOR i IN N_bits TO N_corr+N_bits-1 LOOP int_3(i) := '0'; END LOOP;
END IF;
-- ACCUMULATION (N_corr+N_bits -> N_corr+N_bits)
int_4 := int_4 + int_3;
-- TRONCATURE (N_corr+N_bits+2 -> N_bits+2)
FOR i IN 0 TO N_bits-1 LOOP int_5(i) := int_4(N_corr+i); END LOOP;
-- ADDITION (N_bits -> N_bits)
int_1 := int_5 + binary_in_1;
binary_out_0 <= binary_in_0;
binary_out_1 <= int_1;
END IF;
END IF;
END PROCESS;

END ARCHITECTURE behaviour_2_channel;

```

```

-----
-----Fin du programme-----
-----

```


Bibliographie

- [1] B. Le, T. Rondeau, J. Reed, and C. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, Vol. 22(No. 6) :69–77, Novembre 2005.
- [2] M. Bellanger. *Traitement numérique du signal : Théorie et pratique*. Dunod, 2002.
- [3] M. Benkais. *Méthodologie de caractérisation des circuits de conversion de données : application aux convertisseurs analogique-numérique à facteur de bruit élevé. Mise en oeuvre dans le système CANTesT*. PhD thesis, Université Bordeaux 1, Mai 1993.
- [4] E. Allier. *Interface analogique numérique asynchrone : une nouvelle classe de convertisseur basés sur la quantification du temps*. PhD thesis, Ecole Nationale Supérieure d'Electronique et de Radio électricité de Grenoble TIMA, Novembre 2003.
- [5] A. Oualkadi. *Analyse comportementale des filtres à capacités commutées pour les radio-communications : Conception d'une nouvelle architecture en technologie BiCMOS 0,35 μ m*. PhD thesis, Université de Poitiers, Décembre 2004.
- [6] T. Lagutere. *Conceptions et modélisations d'oscillateurs et de leurs boucles à verrouillage de phase associées pour des applications de radiocommunications mobiles professionnelles*. PhD thesis, Ecole Supérieure d'Ingénieur de Poitiers, Novembre 2005.
- [7] Dj. Haddadi. *Test dynamique des convertisseurs analogique-numérique rapides : caractérisation par analyse temporelle, test industriel et considération instrumentales*. PhD thesis, Université Bordeaux 1, Décembre 2002.
- [8] D. Dallet. *Contribution à la caractérisation des convertisseurs analogique-numérique :évaluation des méthodes et mise en oeuvre de nouveaux procédés*. PhD thesis, Université Bordeaux 1, Janvier 1995.
- [9] C. Recoquillon. *Contribution au dessin et à l'intégration d'un échantillonneur/démultipléxeur à très haut débit pour le projet ALMA*. PhD thesis, Université Bordeaux 1, Novembre 2005.
- [10] M. Comte. *Etude des Corrélations entre Paramètres Statiques et Dynamiques des CAN en vue d'optimiser leur Flot de Test*. PhD thesis, Université Montpellier 2, Juillet 2003.

- [11] IEEE Standard. Ieee standrd for terminology and test methods for analog to digital converters. In *IEEE Std. 1241*, 2000.
- [12] H. Petit. *Simulations comportementales pour la synthèse des convertisseurs analogique numérique CMOS rapide*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Octobre 2004.
- [13] H. Walden. Analog to digital converter survey and analysis. *IEEE Journal on Selected Area in Communications*, Vol. 17(No. 4) :539–550, Avril 1999.
- [14] S. Rapuano. Figure of merit for analog-to-digital converters : Analytic comparaison of international standards. In *IEEE Instrumentation and Measurement Technology Conference*, pages 134–139, Avril 2006.
- [15] Ph. Corvisier. Les convertisseurs analogique numérique rapide. *Electronique Internationale*, Vol. 153 :68–85, Décembre 2004.
- [16] Ch. Rebai. *Contribution à la Caractérisation des Convertisseurs Analogiques Numériques haute performances : Mise en oeuvre de nouveaux systèmes de traitement du signal pour le test in-situ*. PhD thesis, Université Bordeaux 1, Décembre 2002.
- [17] M. Holdway. Le can delta-sigma revoit sa conception et gagne en rapidité. *Electronique*, Vol. 164 :40–43, Décembre 2005.
- [18] M. Jridi. *Rapport de DEA : Implémentation d'un oscillateur numérique à base de modulateur Sigma Delta pour le test in-situ*. Université Bordeaux 1, juin juin 2004.
- [19] W. Black and D. Hodges. Time interleaved data arrays. *IEEE Journal of Solid State Circuit*, SC15 :1022–1029, Décembre 1980.
- [20] Hewlett Packard. Hp54720, hp54722. <http://www.hp.com.fr>.
- [21] Agilent. Agilent. www.agilent.com/labs.
- [22] Atmel. At84ad001. www.atmel.com/dyn/products.
- [23] J. Candy. *Signal Processing : the modern approach*. New York McGraw-Hill, 1988.
- [24] J. Whitaker. *Signal Measurement, Analysis and Testing*. Boca Raton : CRC, 2000.
- [25] J. Simoes, J. Lardeck, and C. Correia. Nonlinearity of a data-acquisition system with interleaving/multiplexing. *IEEE Transaction on Instrumentation and Measurement*, Vol. 46, No. 6 :1274–1279, Décembre 1997.
- [26] J. Simoes, J. Lardeck, and C. Correia. A/d converter performance analysis by a frequency domain approach. *IEEE Transaction on Instrumentation and Measurement*, Vol. 41 :834–839, Décembre 1992.

- [27] J. B. Simoes, J. Landeck, and C. Correira. Nonlinearity of a data-acquisition system with interleaving/multiplexing. *IEEE Transaction in Instrumentation and Measurement*, Vol. 46(No. 6) :1274–1279, Décembre 1997.
- [28] C. Vogel and G. Kubin. Analysis and compensation of nonlinearity mismatches in time interleaved adc arrays. In *IEEE International Symposium on Circuits and Systems*, 2004.
- [29] N. Kurosawa, H. Kobayachi, K. Maryouma, H. Sugawara, and K. Kobayashi. Explicit analysis of channel mismatch effects in time interleaved adc systems. *IEEE Transaction on circuit and systems I : Fundamental theory and application*, Vol. 84, No. 3 :261–271, Mars 2001.
- [30] G. Leger, E. Peralias, A. Rueda, and J. Huertas. Impact of random channel mismatch on the snr and sfdr of time interleaved adcs. *IEEE Transaction on circuit and systems : Regular paper*, Vol. 51(No. 1) :140–150, Janvier 2004.
- [31] H. Jin and E. Lee. A digital background calibration technique for minimizing timing error effects in time interleaved adc's. *IEEE Transaction on Cicuits and Systems*, Vol. 47(No. 5) :603–613, Juillet 2000.
- [32] D. CAMARERO DE LA ROSA. *Calibrage mixte du décalage de l'horloge dans les convertisseurs analogique numérique à entrelacement temporel*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Juin 2007.
- [33] V. T. Nguyen. *Modulator Sigma Delta Passe-Haut et son application au convertisseur Sigma Delta a l'entrelacement temporel*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Juillet 2004.
- [34] U. Eduri. *Online calibration of Nyquist rate Analog to Digital Converters*. PhD thesis, University of Texas at Dallas, Mai 2005.
- [35] M. Jridi, R. Shirakawa, and D. Dallet. Aperture jitter and timing skew analyses in adc structures. In *14th Imeko Symposium on new technologies in measurement and instrumentation and 10th Workshop on ADC modelling and testing*, Septembre 2005.
- [36] Matlab 7.0. Release 14. <http://www.mathworks.com/>.
- [37] J. Pareira, P. Sgirao, and A. Cruz Serra. An fft based method to evaluate and compensate gain and offset errors of interleaved adc systems. *IEEE Transaction on Instrumentation and Measurement*, Vol. 53(No. 1) :423–430, Avril 2004.
- [38] IEEE Std 1076.1-1999. Ieee standard vhdl analog and mixed-signal extensions. New-York 1999.
- [39] M. Tamba, A. Shimizu, H. Munakata, and T. Komuro. A method to improve sfdr with random interleaved sampling method. In *International Test Conference*, 2001.

- [40] S. Jamal, D. Fu, P. Hurst, and S. Lewis. A 10b 120msample/s time-interleaved analog-to-digital converter with digital background calibration. *Solid-State Circuits Conference. Digest of Technical Papers. ISSCC*, Vol. 1 :172–457, 3-7 Février 2002.
- [41] S. Jamal, D. Fu, P. Singh, P. Hurst, and S. Lewis. Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter. *IEEE Transactions on Circuits and Systems I : Regular Papers*, Vol. 37(No. 12) :1618–1627, Décembre 2002.
- [42] S. Jamal, D. Fu, M. Singh, P. Hurst, and S. Lewis. Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter. *IEEE Transaction on Circuits and Systems, Regular papers*, Vol. 51(No. 1) :130–139, Janvier 2004.
- [43] D. Fu, K. Dyer, S. Lewis, and P. Hurst. A digital background calibration technique for time interleaved analog to digital converters. *IEEE journal of Solid-States Circuits*, Vol. 33(No. 12) :1904–1911, Décembre 2003.
- [44] F. Dyer, D. Fu, S. Lewis, and P. Hurst. Analog background calibration of 10-b 40ms/s parallel pipelined adc. *International Solid-State Circuit Conference*, Vol. 7 :142–143, Février 1998.
- [45] R. Jewett, K. Poulton, K. Hsieh, and J. Doernberg. A 12b 128msample/s adc with 0.5 lsb dnl. *International Solid-State Circuit Conference*, Issue 6-8 :138–139, Février 1997.
- [46] M. Jridi, L. Bossuet, B. Le Gal, and D. Dallet. An offset and gain calibration method for time-interleaved analog to digital converters. In *IEEE International Conference on Electronics, Circuits and Systems*, Décembre 2006.
- [47] F. Michaut. *Méthodes adaptatives pour le signal : outils mathématiques et mise en oeuvre des algorithmes*. Hermes, juin 992.
- [48] S. Haykin. *Adaptive filter theory, third edition*. Prentice hall, 1996.
- [49] C. Lang. Least-squares design iir filter with prescribed magnitude and phase responses and a pole radius constraint. *IEEE Transaction on Signal Processing*, Vol. 48(No. 11) :3109–3121, Novembre 2000.
- [50] A. Krukowski and I. Kale. Almost linear-phase polyphase iir lowpass/highpass filter approach. In *International Symposium in Signal Processing and its Applications*, volume Vol. 2, Août 1999.
- [51] K. Grati. *Architecture d'un récepteur radio multistandard à sélection numérique des canaux*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Juin 2005.
- [52] F. Michaut and M. Bellanger. *Filtrage adaptatif : Théorie et algorithmes*. Lavoisier, 2005.
- [53] P. Viot. *Méthodes d'analyse numérique*. Laboratoire de Physique Théorique des Liquides, 9 Janvier 2007.

- [54] J. Marcelin. *Méthodes numériques pour l'ingénieur*. Hermes, 2001.
- [55] V. Martin. *Contribution des filtre LPTV et des techniques d'interpolation au tatouage numérique*. PhD thesis, Institut national polytechnique de Toulouse, Novembre 2006.
- [56] P. Thévenaz, T. Blu, and M. Unser. *Image interpolation and resampling*. Handbook of medical imaging processing and analysis, 2000.
- [57] M. Benkais, S. Le Masson, and P. Marchegay. A/d converter characterization by spectral analysis in dual-tone mode. *IEEE Transaction in Instrumentation and Measurement*, Vol. 44(No. 5) :940–944, Octobre 1995.
- [58] G. Long, F. Ling, and G. Proakis. The lms algorithm with delayed coefficient adaptation. *IEEE Transaction on Acoustics Speech and Signal Processing*, Vol. 37(No. 9) :1397–1405, Septembre 1989.
- [59] D. Boudreau and P. Kabal. Joint time-delay estimation and adaptive recursive least squares filtering. *IEEE Transaction on Signal Processing*, Vol. 41(No. 2) :592–601, Février 1993.
- [60] D. Meyer and P. Agrawal. Vectorization of the dlms transversal adaptive filter. *IEEE Transaction on Signal Processing*, Vol. 42(No. 11) :3237–3240, Novembre 1994.
- [61] F. Laichi, T. Aboulnasr, and W. Steenaart. Effect of delay on the performance of the leaky lms adaptive algorithm. *IEEE Transaction on Signal Processing*, Vol. 45(No. 3) :775–780, Mars 1997.
- [62] S. Douglas, Q. Zhu, and K. Smith. A pipelined lms adaptive fir filter architecture without adaptation delay. *IEEE Transaction on Signal Processing*, Vol. 46(No. 3) :775–780, Mars 1998.
- [63] O. Tobias and R. Seara. Leaky delayed lms algorithm : Stochastic analysis for gaussian data and delay modeling error. *IEEE Transaction On Signal Processing*, Vol. 52(No. 6) :1596–1606, Juin 2004.
- [64] M. Meyer and D. Agrawal. A high sampling rate delayed lms filter architecture. *IEEE Transaction in Circuits and Systems II Analog and Digital Processing*, Vol. 40(No. 11) :727–729, Novembre 1993.
- [65] C. Kim, H. Soeleman, and K. Roy. Ultra-low-power dlms adaptive filter for hearing aid applications. *IEEE Transaction on Very Large scale Integration (VLSI) Systems*, Vol. 11(No. 6) :1058–1067, Décembre 2003.
- [66] R. Perry, D. Bull, and A. Nix. Pipelined dfe architectures using delayed coefficient adaptation. *IEEE Transaction on Circuits and Systems II Analog and Digital Processing*, Vol. 45(No. 7) :868–873, Juillet 1998.

- [67] S. Ahn and P. Votz. Convergence of the delayed normalized lms with decreasing step size. *IEEE Transaction on Signal Processing*, Vol. 44(No. 12) :3008–3017, Décembre 1996.
- [68] C. Vogel and H. Johansson. Time-interleaved analog to digital converters : status and future direction. In *IEEE International Symposium on Circuits and Systems*, pages 3386–3389, Décembre 2006.
- [69] J. Hamon. *Méthodes et outils de la conception amont pour les systèmes et les microsystèmes*. PhD thesis, Institut National Polytechnique de Toulouse, Février 2005.
- [70] A. Vachoux. *Modélisation de Systèmes Intégrés Analogiques et Mixtes*. Laboratoire de Systèmes Microélectroniques EPFL, Février 2002.
- [71] ModelSim. Mentor graphics. <http://www.mentor.com/products/fv/digital>.
- [72] NC Sim. Cadence. www.eecs.uc.edu/cadence/docs/vhdlsimcadencetutorial.pdf.
- [73] SPICE. Microsim. <http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/>.
- [74] Eldo. Mentor. <http://www.mentor.com/products/ic>.
- [75] Spectre. Cadence. http://www.cadence.com/products/custom_ic/spectre/index.aspx.
- [76] Mentor Graphics. *ADVanceMS reference manual*. Mentor Graphics, 2000.
- [77] S. Renaud. *Contribution à la caractérisation des circuits de conversion analogique-numérique. Conception et réalisation d'un système d'évaluation dynamique de ces dispositifs*. PhD thesis, Université de Bordeaux 1, Janvier 1990.
- [78] M. Jridi, G. Monnerie, L. Bossuet, and D. Dallet. Two time-interleaved adc channel structure : analysis and modeling. In *IEEE Instrumentation and Measurement Technology Conference*, pages 781–785, Avril 2006.
- [79] L. Bossuet. *Exploration de l'Espace de Conception des Architectures Reconfigurables*. PhD thesis, Université de Bretagne Sud, 10 Septembre 2004.
- [80] N. Tredennick and B. Shimamoto. The rise of reconfigurable systems. In *Engineering of Reconfigurable Systems and Application*, Juin 2003.
- [81] P. Chow, S. Seo, J. Rose, G. Paez-Monzon, and I. Rahardja. The design of an sram-based field-programmable gate array - part i : Circuit design and layout. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 7(No. 2) :321–330, Juin 1999.
- [82] Virtex. *Platform FPGAs Technical Document*. Xilinx, Août 2003.
- [83] Analog Device. Ad9218. <http://www.analog.com/UploadedFiles/DataSheets/AD9218.pdf>.
- [84] M. Jridi, D. Dallet, Ch. Rebai, and P. Marchegay. Digital sigma delta oscillator : Design consideration. In *Design of Circuits and Integrated Systems*, Décembre 2004.

- [85] Y. Hervé. *Un langage pour l'électronique du futur*. Université Louis Pasteur de Strasbourg, 2002.
- [86] G. Monnerie. *Etude et modélisation de sources de bruit dans des structure à temps discret*. PhD thesis, Université de Bordeaux 1, Juillet 2005.

Résumé

Dans un contexte où les systèmes communicants fleurissent, les Convertisseurs Analogique Numérique CAN doivent suivre les demandes des nouveaux standards de télécommunications. Un convertisseur seul, ne peut pas allier rapidité, précision et faible consommation de puissance. Dans le cadre de nos travaux, nous sommes intéressés à une structure prometteuse de CAN basée sur l'entrelacement temporel de plusieurs convertisseurs, TIADC. Le taux d'échantillonnage augmente proportionnellement avec le nombre de CAN mais des problèmes de disparité entre les différents CAN réduisent la résolution effective du TIADC. Dans ce mémoire, nous avons contribué à l'étude de ces convertisseurs, notamment aux pertes engendrées par les disparités entre les différents convertisseurs. La structure du TIADC a été modélisée dans un environnement de description matérielle. Plusieurs solutions de calibrations existantes ont été simulées afin de vérifier leur fonctionnement et de pouvoir proposer deux méthodes de correction. Une première méthode en différé visant le domaine de l'instrumentation et une seconde, en ligne visant des applications de télécommunications. La première méthode a été vérifiée par des données expérimentales, la seconde était implémentée dans un FPGA et vérifiée par des tests et des mesures.

Mots clés : CAN, méthodes de calibration, implantation matérielle, filtrage adaptatif.

Abstract

To comply with new telecommunication standard requirements, Analog to Digital Converter ADC should provide high sample rate, high resolution and low power consumption. With today's technologies, only one converter cannot achieve these requirements. To cope with this problem, one solution consists in parallelizing existing ADC cores on a die to increase the sampling rate for the same resolution and power consumption. The resulting system is called Time Interleaved Analog to Digital Converter TIADC. However, TIADC systems introduce new set of problem : indeed, each converter is characterized by its proper errors, mismatches between ADC cause undesirable spurs. The main thesis contribution is about TIADC spectral parameter loss study caused by mismatches effect, TIADC modelling and material description. A special attention is given to TIADC compensation methods by proposing two solutions : the first one is an offline method used for measurement applications. The second is an online method based on adaptive filtering for telecommunications applications. The first method is verified by experimental data, the second one is implemented into FPGA and verified by tests and measures.

Key words : ADC, calibration method, material implementation, adaptive filtering