



HAL
open science

Etude et compensation des non-linéarités de convertisseur analogique numérique utilisant une architecture à repliement et interpolation

Vincent Fresnaud

► **To cite this version:**

Vincent Fresnaud. Etude et compensation des non-linéarités de convertisseur analogique numérique utilisant une architecture à repliement et interpolation. Micro et nanotechnologies/Microélectronique. Université Sciences et Technologies - Bordeaux I, 2008. Français. NNT : . tel-00288869

HAL Id: tel-00288869

<https://theses.hal.science/tel-00288869>

Submitted on 18 Jun 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 : 3579

THÈSE

présentée à

L'UNIVERSITÉ DE BORDEAUX I

ECOLE DOCTORALE DE SCIENCES PHYSIQUES ET DE L'INGENIEUR

par Vincent FRESNAUD

POUR OBTENIR LE GRADE DE
DOCTEUR

SPECIALITE : ELECTRONIQUE

ETUDE ET COMPENSATION DES NON-LINÉARITÉS DE CONVERTISSEUR ANALOGIQUE
NUMÉRIQUE UTILISANT UNE ARCHITECTURE À REPLIEMENT ET INTERPOLATION

Soutenue le 07 Avril 2008

Jury composé de :

GARDA	Patrick	Professeur, Université Pierre et Marie Curie/LIS, Paris	Rapporteur
MIR	Salvador	Directeur de Recherche, TIMA, Grenoble	Rapporteur
DALLET	Dominique	Professeur, ENSEIRB/IMS, Bordeaux	Directeur de thèse
RENAUD	Sylvie	Professeur, ENSEIRB/IMS, Bordeaux	Présidente
AGNUS	Benoît	Industrial manager, NXP Semiconductors, Caen	Examineur
BERNARD	Serge	Chargé de recherche CNRS/LIRMM, Montpellier	Examineur
BOSSUET	Lilian	Maître de conférences, ENSEIRB/IMS, Bordeaux	Examineur

Remerciements

Ce document présente les travaux que j'ai effectués durant trois ans au sein de NXP Semiconductors et je remercie les personnes de cette entreprise qui m'ont permis de les réaliser dans de bonnes conditions et en particulier M. Benoît AGNUS, M. Philippe GANDY et M. Philippe CAUVET. Je remercie aussi chaleureusement tout le personnel de cette entreprise qui m'a très bien accueilli.

Je remercie M. Salvador MIR et M. Patrick GARDA d'avoir accepté d'être les rapporteurs de cette thèse et de faire partie du jury lors de la soutenance. Je remercie également Mme Sylvie RENAUD qui a acceptée de présider le jury.

Je remercie particulièrement mes encadrants universitaires M. Dominique DALLET, M. Lilian BOS-SUET et M. Serge BERNARD pour m'avoir accompagné tout au long de ce parcours initiatique malgré la distance géographique qui nous séparait. Qu'ils reçoivent à travers ces lignes toute mon amitié et mon respect.

Je remercie également le personnel des Laboratoires IMS de Bordeaux et LIRMM de Montpellier, et en particuliers M. Guillaume MONNERIE et Mlle Marianne COMTE pour leur aide ponctuelle mais pertinente.

Je remercie toutes les personnes du service PL Data Converters de NXP qui m'ont permis de mener à bien mes travaux, et en particuliers M. Philippe VINCENT, M. Gilles JENVRIN, M. Stéphane JOUIN, M. Christophe ERDMANN, M. Fabrice ROUILLE. Sans leur clairvoyance et leur aide précieuse, ces travaux n'auraient pu être menés à terme.

Je remercie tout les thésards de NXP au travers de M. Jean-Noel PALAZIN, pour la communauté qui nous relie et nous permet d'échanger sur des sujets aussi divers que passionnants. Ce mélange des genres est un terrain fertile au travail en équipe.

Je remercie la folle équipe des Jeudy's : Vince K & Anne-Laure & Mayeul, Vince G & Magali, Guillaume & Nath, Jay & Amandine, Arnaud & Arnaud, Seb & Lilou, Aurélien, Élise, Estelle & David pour les débats contradictoires du Jeudi soir sur le choix du pichet à commander, ainsi que pour leur amitié.

Je remercie Hoegaardenement Jérôme & Titi pour leur convivialité et leur générosité à toute épreuve.

Un grand MERCI aux "Pétas" pour leur profonde amitié et leur support quotidien lors de ces dernières années et en particuliers lors des derniers mois. Encore une fois, merci les filles (vraies ou fausses :)) .

Un grand MERCI aux Renés : Rhum & Nirm & Harold, Sulli & Pépette & (?) et Nico, pour leur amitié sans faille au travers de toutes ces années, et ce malgré la distance qui nous sépare. Merci !

Un MERCI mégalithique à la communauté des MENHIRS : Jon & Mathilde, SoK & Muriel, Samuel & Sandrine, Laaapin & Thao, Kiri, Olivier & Bénédicte, Titi & Valérie, Solène & Nico & Lucas. Depuis plus de 10ans, votre présence quasi quotidienne nous laisse entrevoir beaucoup d'autres grands moments :) .

Un énorme MERCI à ma famille pour m'avoir accompagné une fois encore sur une étape importante de ma vie. Qu'ils reçoivent à travers ces mots tout mon respect et mon amour, et plus que tout, toute ma gratitude. Et en route pour la prochaine étape !

Finalement, un MERCI qui repousse les frontières de l'univers à mes Parents et ma Mamie pour leur infinie sagesse et leur présence à toute épreuve, et ce, depuis ma naissance. Je souhaite à tout le monde d'avoir des proches tels que vous. Merci !

Table des matières

Table des matières	6
Introduction	11
1 Contexte de l'étude	15
1.1 Introduction	15
1.2 Échantillonneur-Bloqueur	16
1.3 La quantification analogique numérique	17
1.4 Les paramètres de performance du CAN	20
1.4.1 Les paramètres statiques	20
1.4.2 Les paramètres dynamiques	21
1.5 Architectures des Convertisseurs Analogique Numérique	23
1.5.1 Architecture Flash	23
1.5.2 Architecture Flash à Interpolation	24
1.5.3 Architecture par partitionnement	25
1.5.4 Architecture à Repliement	27
1.5.5 Limitations de l'architecture à repliement et à interpolation	33
1.6 Nouvelle stratégie de conception et conclusion	35
2 Méthodes de Compensation	39
2.1 Introduction	39
2.2 Méthodes basées sur l'utilisation d'une table de correspondance	39
2.2.1 Correction statique	40
2.2.2 Correction dynamique	41
2.2.3 Remplissage des Look-Up Table	45

TABLE DES MATIÈRES

2.2.4	Bilan de la compensation par table de correspondance	45
2.3	Méthodes basées sur la modélisation des Convertisseurs	46
2.3.1	Les séries de Volterra	46
2.3.2	Identification du modèle	47
2.3.3	Inversion et compensation	48
2.4	Méthodes de compensation interne	48
2.4.1	La technique du dither	48
2.4.2	Calibration des capacités pour un convertisseur Pipeline	49
2.4.3	Correction numérique adaptative dans un convertisseur $\Sigma\Delta$	50
2.5	Conclusion	52
3	Extraction des non-linéarités	53
3.1	Introduction	53
3.2	Extraction basée sur une approche statistique	53
3.2.1	Méthode classique de test par histogramme	54
3.2.2	Utilisation en vue de la compensation	55
3.3	Extraction à partir du spectre de sortie du CAN	56
3.3.1	Modélisation des non-linéarités par un polynôme	57
3.3.2	Modélisation des non-linéarités par série de Fourier	60
3.4	Conclusion	65
4	Etude de l'efficacité de la compensation <i>offline</i>	67
4.1	Introduction	67
4.2	Véhicule de Test	67
4.2.1	Le convertisseur AD_{DEMO}	68
4.2.2	Spécifications et configuration du FPGA	70
4.3	Etude de la robustesse de la méthode de compensation	72
4.3.1	Comparaison des méthodes de calcul de la NLI	73
4.3.2	Rappels sur l'échantillonnage et le repliement spectral	74
4.3.3	Efficacité de la compensation selon la variation des fréquences d'utilisation	77
4.3.4	Efficacité de la compensation en fonction de la fréquence d'échantillonnage	80
4.4	Conclusion	85

5 Autocompensation embarquée	87
5.1 Présentation de la compensation embarquée	87
5.1.1 Unité de commande et de calcul	88
5.2 Réduction de la complexité de la méthode spectrale	90
5.2.1 Modélisation de la courbe des non-linéarités pour une architecture à repliement et interpolation	90
5.2.2 Impact de l'architecture sur la matrice $T_{K_{MAX}}$	93
5.2.3 Impact de l'architecture sur le choix des Harmoniques du signal converti	94
5.2.4 Conclusion	97
5.3 Calcul des harmoniques In Situ	98
5.3.1 Notations	98
5.3.2 La transformée de Fourier	99
5.3.3 L'algorithme de Goertzel	100
5.3.4 Notre solution : l'algorithme LoCHE	102
5.3.5 Implémentation et comparaison des techniques	108
5.3.6 Conclusion	111
5.4 Conclusion	112
Conclusion générale	113
Bibliographie	117
Annexes	123
Résumé/Abstract	131

Glossaire

ATE :	automatic test equipment
BIST :	built-in-self-test
CAN :	convertisseur analogique numérique
CEM :	compatibilité électro-magnétique
CNA :	convertisseur numérique analogique
DFT :	discrete fourier transform
E/B :	echantillonneur bloqueur
ENOB :	effective number of bits
FFT :	fast fourier transform
FIFO :	first in first out
FIR :	finite impulse response
FPGA :	field programmable gate array
LoCHE :	low cost harmonics estimation
LUT :	look-up table
LSB :	less significant bit
MSB :	most significant bit
NLD :	non linéarité différentielle
NLI :	non linéarité intégrale
PE :	pleine échelle
RTC :	réseau téléphonique commuté
SAW :	surface acoustic wave
SDR :	software defined radio
SFDR :	spurious free dynamic range
SINAD :	signal to noise and distortions ratio
SNR :	signal to noise ratio
SVD :	singular value decomposition
THD :	total harmonic distortion to signal ratio
VHDL :	Very high speed integrated circuit Hardware Description Language

Introduction

L'avènement de la société numérique constitue une révolution technologique dont les implications économiques, sociales, mais aussi culturelles, placent l'homme au coeur d'un réseau communiquant de plus en plus complexe.

Le principe de base de cette communication consiste en un échange d'informations entre un émetteur et un récepteur par l'intermédiaire d'un signal. Quand deux personnes se parlent, ils constituent les extrémités de la chaîne de communication (émetteur/récepteur). Le son constitue quant à lui, le signal contenant l'information à échanger. Pour se transporter de l'émetteur au récepteur, le signal a besoin d'un support, qui dans notre exemple s'avère être tout simplement l'air. Ce support se nomme habituellement canal de transmission et possède ses propriétés propres. Par exemple, l'air atténue l'onde sonore, ce qui implique une distance critique à partir de laquelle le récepteur ne peut plus entendre le message de l'émetteur. Pour palier à ce phénomène, l'homme utilise des interfaces qui transforment le signal en un autre signal qui peut être véhiculé au sein d'un canal de transmission plus adapté aux grandes distances. L'invention du téléphone par Graham Bell constitue la première étape de cette révolution. La voix de l'émetteur est transformée en un signal électrique qui peut être transmis au récepteur à l'aide d'un câble conducteur. Le récepteur possède également une interface qui lui permet de reconstituer le signal original. La nouvelle chaîne s'exprime désormais comme indiquée sur la figure 1, à savoir : émetteur - interface - canal - interface - récepteur.

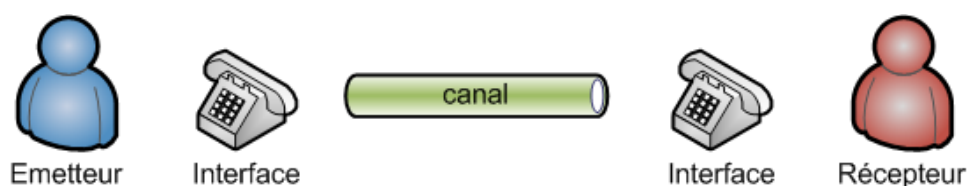


FIG. 1 – Chaîne de communication mono-canal

Cette structure, bien que fonctionnelle, possède ses limites. Par exemple, elle implique qu'entre chaque émetteur et récepteur différents, un canal doit être présent. Il n'est pas envisageable d'avoir autant de canaux que de connexions possibles entre l'ensemble des émetteurs et des récepteurs. Les interfaces ont donc été modifiées afin de pouvoir utiliser un même canal pour un ensemble d'émetteurs/récepteurs, comme indiqué sur la figure 2. C'est le cas du réseau téléphonique commuté (RTC) dont le principe est toujours utilisé de nos jours.

Depuis quelques années, l'introduction de nouveaux services (téléphonie mobile, télévision numérique,

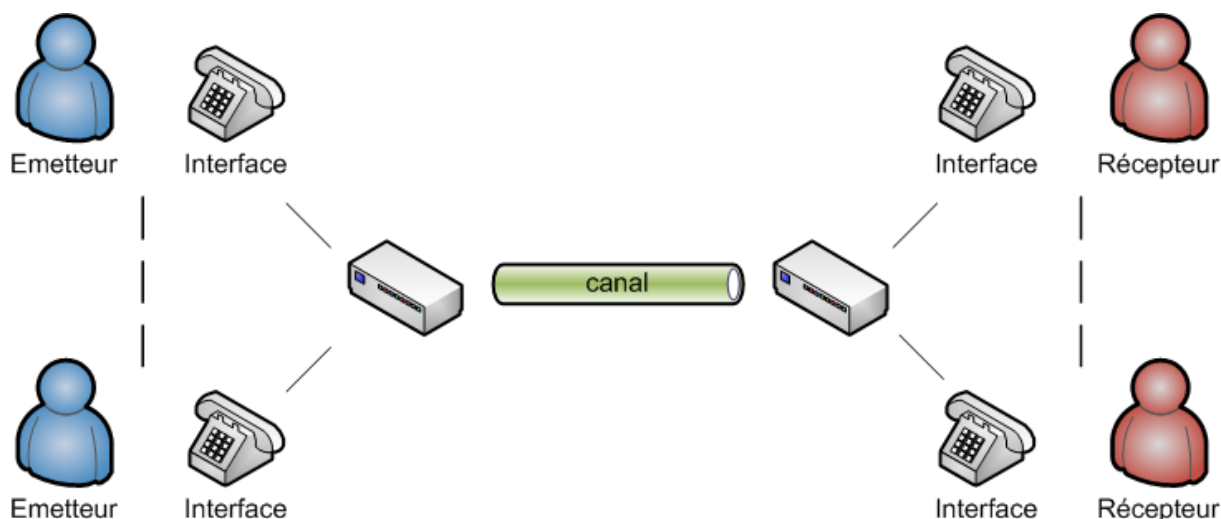


FIG. 2 – Chaîne de communication multi-utilisateurs

...), la convergence des différents supports (filaire, radio,...), et la volonté d'améliorer et de rentabiliser l'utilisation des liaisons, sont autant de raisons qui poussent les fournisseurs de services et les opérateurs de télécommunications à numériser leurs réseaux. En introduisant des interfaces qui transforment les signaux analogiques (voix, images par exemple) en signaux numériques (suite d'information binaire), ces acteurs industriels s'assurent une souplesse de programmation (reprogrammable selon les nouveaux standards), une reproductibilité (la dépendance de l'environnement est moindre (CEM,...)) et un niveau de qualité (codes correcteurs d'erreur, réémissions,...) beaucoup plus élevés qu'avec les communications analogiques. La figure 3 représente la chaîne de communication numérique.

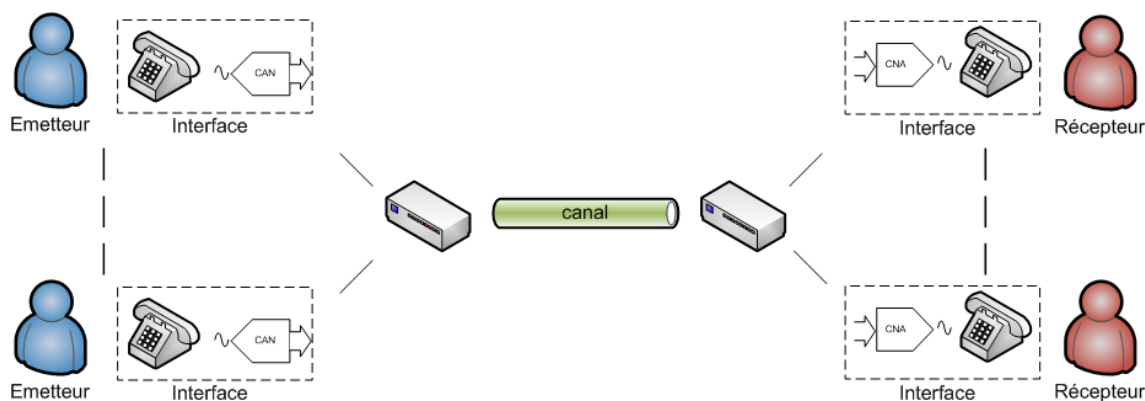


FIG. 3 – Chaîne de communication numérique

Le maillon critique de cette nouvelle chaîne est le composant de l'interface permettant la conversion des signaux physiques (de nature analogique), en signaux numériques. Pour l'interface émettrice, il s'agit du Convertisseur Analogique Numérique (CAN), tandis que pour l'interface réceptrice, il s'agit du Convertisseur Numérique Analogique (CNA). Poussés par cette numérisation des systèmes de télécommunications, les convertisseurs analogiques numériques et numériques analogiques, se font toujours plus

rapides, plus précis, plus intégrés avec une réduction de la consommation en puissance et en énergie. Les technologies et architectures actuelles permettent déjà d'atteindre des performances appréciables, mais chaque application requiert son propre convertisseur selon les caractéristiques de fonctionnement des différents standards (fréquence d'échantillonnage, bande de fréquence d'entrée, résolution, ...). Depuis quelques années, la tendance est à la convergence des différents standards au sein d'un même système : c'est la Software Defined Radio (SDR). La SDR permet avec un même appareil reprogrammable de décoder l'ensemble des protocoles de communications. Pour ce faire, l'appareil peut embarquer un convertisseur spécifique pour chaque protocole, mais l'encombrement physique de cet assemblage et la consommation totale ne sont pas adaptés aux applications mobiles actuelles. L'idéal est de posséder un seul convertisseur dont les caractéristiques conviennent à l'ensemble des standards existants et à venir. Cela implique la création d'un convertisseur à fréquence d'échantillonnage très élevée (plusieurs Giga-Hertz), et de grande résolution (au moins 18bits). De plus, la bande des fréquences d'entrée se doit d'être très large (environ 2MHz à 4GHz). De tels convertisseurs semblent pour le moment hors d'atteinte. De nombreuses recherches sont en cours pour améliorer les composants actuels en proposant de nouvelles architectures et de nouveaux procédés de fabrication. Cette évolution est longue et doit se faire étape par étape. Cependant, il est possible d'utiliser des méthodes permettant de compenser les lacunes d'un composant donné avant de franchir l'étape suivante. Ces méthodes de compensation permettent de repousser les limites du composant en attendant la maturité de la génération suivante. Elles permettent également de mieux comprendre les défauts actuels et d'orienter les concepteurs vers des pistes prometteuses pour leurs recherches. C'est dans ce contexte que les travaux de cette thèse ont été menés.

Le premier chapitre de ce mémoire est consacré à la problématique générale liée à la conception des convertisseurs analogiques numériques. Après avoir exposé les principes de base de la conversion, quelques architectures sont présentées afin de mieux discerner les challenges de conception et de caractérisation de ces composants. Nous présentons également la stratégie de conception qui vise à relaxer les paramètres sensibles du convertisseur afin d'accroître sa vitesse de fonctionnement. Cette relaxation implique une augmentation des non-linéarités qu'il faut pouvoir compenser par la suite.

Le second chapitre présente un état de l'art des méthodes de compensation. Au travers des techniques par table de correspondance, par modèle analytique et celles liées à l'architecture interne, nous proposons de nous focaliser sur l'une d'entre elle afin de réaliser une compensation simple et efficace.

Le troisième chapitre s'articule autour des méthodes d'extraction de la courbe des non-linéarités du convertisseur. Après avoir exposé la méthode statistique classique, nous proposons d'utiliser et d'optimiser une méthode analytique basée sur une étude fréquentielle du signal converti. Ce nouvel axe nous permet de réduire les besoins en ressources matérielles et logicielles pour effectuer la compensation.

Le quatrième chapitre se consacre à l'étude de l'efficacité de la technique de compensation utilisée. Au travers d'un convertisseur spécialement modifié pour notre étude et d'un composant logique reprogrammable, nous mettons en oeuvre une série d'expérimentations afin de discriminer les paramètres sensibles de la méthode de compensation.

Le cinquième chapitre ouvre la voie à la compensation embarquée. Afin d'utiliser notre méthode d'extraction des non-linéarités à partir du spectre du signal converti, nous proposons d'étudier l'implémentation physique de l'extraction des paramètres spectraux. Deux techniques classiques sont d'abord pré-

TABLE DES MATIÈRES

sentées et discutées, puis une nouvelle approche est exposée. Ce nouvel algorithme propose d'estimer la valeur de quelques raies spectrales à partir de simple additionneurs et soustracteurs, réduisant ainsi la complexité de calcul à embarquer.

Enfin, le dernier chapitre conclut ce manuscrit en résumant les apports de notre approche et les perspectives qui s'en dégagent.

Chapitre 1

Contexte de l'étude

Le premier chapitre de ce mémoire est consacré à la problématique générale liée à la conception d'un convertisseur analogique numérique. Après avoir exposé les principes de base de la conversion, quelques architectures sont présentées afin de mieux discerner les challenges de conception et de caractérisation de ces composants.

1.1 Introduction

La Conversion Analogique Numérique a pour fonction de transformer un signal d'entrée analogique, généralement un signal électrique, en un code binaire sur plusieurs bits.

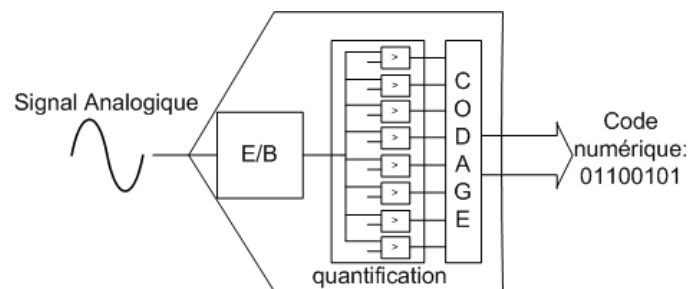


FIG. 1.1 – Convertisseur Analogique Numérique

Cette conversion s'effectue à travers différentes étapes illustrées en figure 1.1. La première étape est la capture du signal d'entrée. Cette capture s'effectue généralement par l'intermédiaire d'un étage dit échantillonneur-bloqueur (E/B). Cette étape est indispensable dans le cas des signaux dont la fréquence de variation est très élevée en regard de la vitesse de fonctionnement de la circuiterie électronique en aval. Son utilisation permet d'augmenter la gamme de fréquence à l'entrée du convertisseur.

La seconde étape est la plus sensible, il s'agit de la quantification de la valeur précédemment capturée. Il existe différentes façons de réaliser cette quantification : on parle alors d'architectures de convertisseurs. Le principe de base reste commun à toutes ces architectures : la valeur capturée est comparée à divers

signaux électriques de références prédéfinis. Cette première représentation est ensuite encodée pour fournir un code de sortie binaire utilisable par d'autres composants.

Les sections suivantes détaillent les différents étages ainsi que les limitations des architectures possibles.

1.2 Échantillonneur-Bloqueur

La première étape de la conversion analogique numérique consiste à capturer la valeur du signal d'entrée pour la présenter à l'entrée des différents comparateurs. Pour que ces derniers puissent fonctionner correctement, il faut que la tension (ou le courant) soit stable pendant le temps d'établissement de la comparaison. La figure 1.2 représente l'évolution temporelle d'un signal basse fréquence. Le grossissement de la figure représente l'évolution du signal pendant sa capture par un comparateur. Le temps d'établissement de la comparaison est représenté par Δt . Dans le cas d'un convertisseur de tension, cet intervalle temporel implique un intervalle de tension Δv qui doit être inférieur à la sensibilité s du comparateur. Dans notre exemple, la variation du signal d'entrée est faible et les comparateurs fournissent une valeur de sortie correcte.

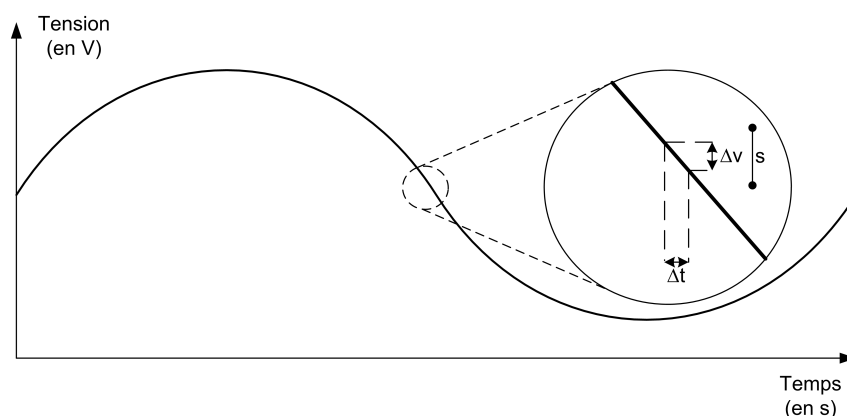


FIG. 1.2 – Capture signal basse fréquence

Pour un signal haute fréquence, la pente du signal est plus importante et l'intervalle de tension Δv parcouru lors du même intervalle Δt peut devenir plus grand que la sensibilité s du comparateur (figure 1.3). Ce dernier ne peut donc pas fournir une valeur de sortie représentative du signal d'entrée.

L'échantillonneur-bloqueur (E/B) a été conçu pour résoudre ce problème. Comme son nom l'indique, il se compose de deux étapes successives, résumées sur la figure 1.4. La première étape consiste en une poursuite du signal d'entrée pendant un temps dit d'échantillonnage. Le passage à la seconde étape se fait de façon synchrone avec l'horloge du convertisseur. Lors de cette seconde étape, on cherche à maintenir (bloquer) la valeur du signal échantillonné. Après un temps d'établissement, la sortie est stable et peut être traitée par les comparateurs.

Les contraintes technologiques de l'E/B sont très fortes. Il faut en effet que l'ensemble soit très réactif : d'une part l'échantillonneur doit pouvoir très rapidement recopier le signal d'entrée quelque soit sa fréquence, et d'autre part, le temps d'établissement du bloqueur doit être le plus court possible afin

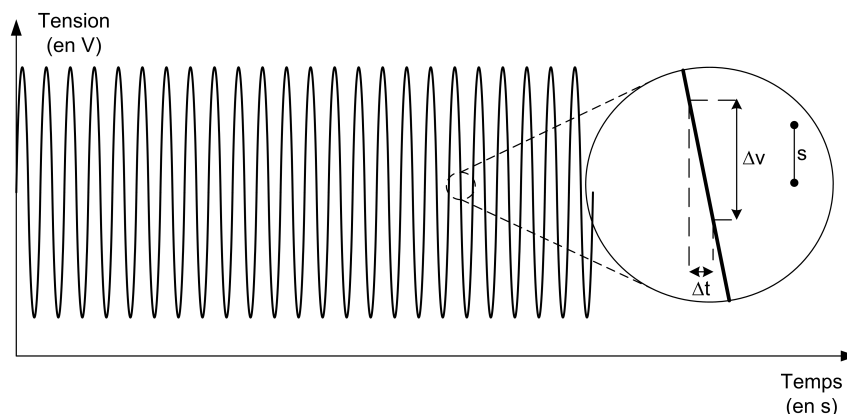


FIG. 1.3 – Capture signal haute fréquence

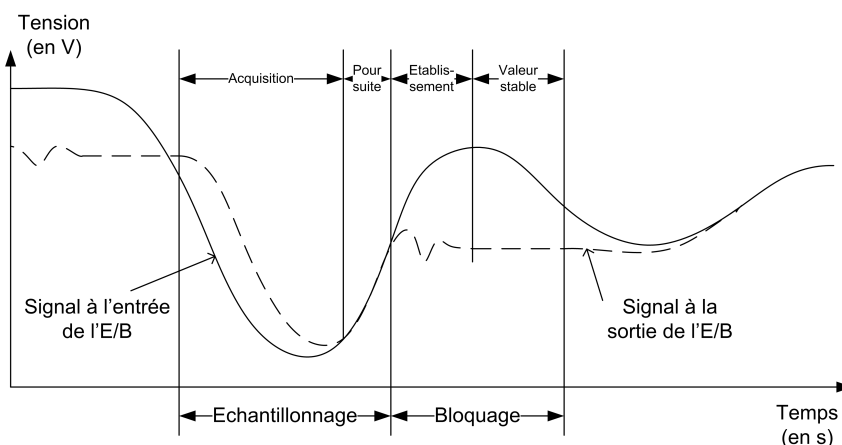


FIG. 1.4 – Fonctionnement temporel de l'échantillonneur-bloqueur

d'augmenter le temps de valeur stable. Plus ces deux organes travaillent vite, plus le convertisseur pourra fonctionner à des fréquences d'horloge élevées. La difficulté de conception de ces deux blocs est un des éléments limitatifs à l'obtention de convertisseurs à haute vitesse.

1.3 La quantification analogique numérique

La quantification consiste en la conversion d'un ensemble continu, donc infini, de valeurs en un ensemble discontinu et fini de valeurs. Dans le cas d'un CAN, cette quantification transforme un signal électrique en un code binaire représenté sur n bits. La figure 1.5 illustre la quantification d'une tension (à gauche de la figure) en un code binaire codé sur $n = 3$ bits (à droite de la figure). Chaque ellipse correspond à un intervalle borné mais contenant un nombre infini de valeurs possibles. La quantification est représentée par les flèches qui, à chaque ellipse associe un code binaire. Les extrémités $]-\infty; V_1]$ et $[V_7; +\infty[$ sont volontairement saturées aux codes extrêmes 000 et 111.

On nomme *tension de transition* le niveau V_i permettant de basculer du code $i - 1$ au code i .

1.3. LA QUANTIFICATION ANALOGIQUE NUMÉRIQUE

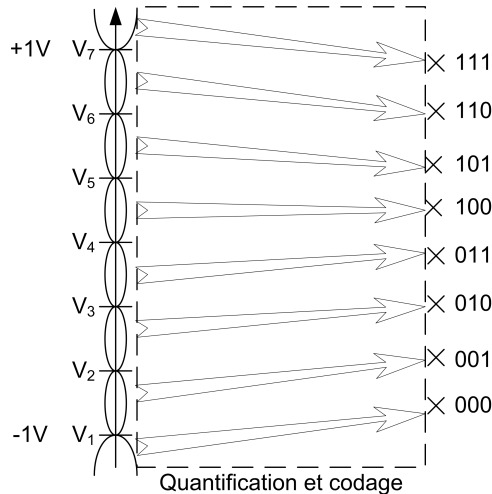


FIG. 1.5 – Principe de la quantification

On définit la fonction de transfert de cette conversion comme étant la fonction qui associe à chaque plage de valeurs i , sa valeur codée en binaire. Ainsi la plage de tension numéro 5, définie par les tensions V_5 et V_6 , est codée par le code binaire 101. Cette fonction de transfert se traduit graphiquement par la figure 1.6(a), sur laquelle on observe la forme caractéristique en escalier. Ce type de convertisseur se nomme convertisseur par troncature qui implique une erreur à moyenne non nulle comme indiquée sur la figure 1.6(a).

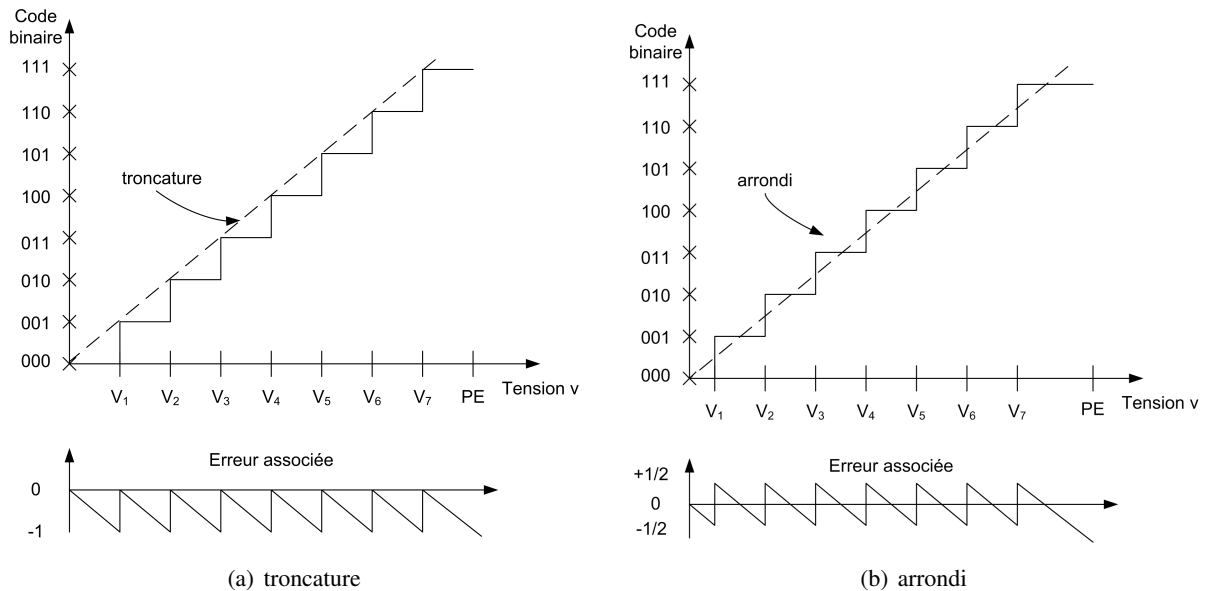


FIG. 1.6 – Fonction de transfert

Pour avoir une moyenne nulle, on préfère procéder par arrondis (figure 1.6(b)).

Pour une quantification uniforme, la distance entre deux tensions de transitions successives est constante. Cette distance se nomme *pas de quantification*, ou *quantum*. Sa valeur dépend du nombre de bits n choisi pour décrire les codes binaires ainsi que de la dynamique en amplitude des signaux d'entrée. Avec n

bits, le cardinal de l'ensemble des valeurs discrètes est de 2^n , on peut donc diviser l'amplitude maximale, nommée *pleine échelle* (PE), en autant de plages. Le pas de quantification q est donc décrit par la formule suivante :

$$q = \frac{PE}{2^n} \quad (1.1)$$

Plus la valeur de n est grande, plus le pas de quantification sera petit, et donc meilleure sera la conversion. C'est pour cette raison que n est appelé *résolution* du convertisseur. Le passage d'un code binaire au suivant se fait en analogique par une variation égale au pas q , ce qui équivaut à associer le pas de quantification q analogique au bit de poids faible du codage binaire : le *LSB* (*Least Significant Bit*).

L'erreur associée à la conversion (figure 1.6) est appelée *erreur de quantification*. Pour un convertisseur idéal, cette erreur est le seul facteur limitatif de la conversion. Toute variation du signal d'entrée inférieure au pas de quantification q , se retrouve noyée dans ce bruit. Pour mesurer la performance du convertisseur, on calcule le rapport signal sur bruit (SNR : *Signal-to-Noise Ratio*), c'est à dire la plus grande distance entre la puissance du signal d'entrée, et la puissance du bruit. Plus cette valeur est grande, plus le convertisseur est à même de reproduire fidèlement le signal d'entrée. Dans [Gra90], le bruit de quantification ϵ est modélisé par un processus aléatoire stationnaire, décorrélé de l'entrée analogique et de lui même, et uniformément réparti. Sous ces hypothèses et dans le cas d'une conversion par arrondi, sa puissance peut être décrite par la formule suivante [Ger78] :

$$\sigma_\epsilon^2 = \frac{q^2}{12} \quad (1.2)$$

Dans le cas d'un signal sinusoïdal parcourant la pleine échelle, la puissance P_S du signal d'entrée s'exprime par l'équation suivante :

$$P_S = \frac{\left(\frac{PE}{2}\right)^2}{2} \quad (1.3)$$

A l'aide de l'équation 1.1, on obtient :

$$P_S = \frac{2^n q^2}{8} \quad (1.4)$$

Ce qui donne le rapport signal sur bruit dû à la quantification SNR_Q suivant :

$$\begin{aligned} SNR_Q &= 10 \log \left(\frac{2^{2n} q^2 \cdot 12}{q^2 \cdot 8} \right) \\ &= 10 \log(3 \times 2^{2n}) \\ &= 6.02n + 1.76 \text{ (dB)} \end{aligned} \quad (1.5)$$

Le terme SNR_Q représente le rapport signal sur bruit maximal que l'on puisse obtenir avec un convertisseur idéal, la limitation provenant uniquement de la quantification. Dans le cas réel, la conception du convertisseur fait intervenir différentes sources d'erreurs qui créent des distorsions sur le signal converti et du bruit supplémentaire. Le rapport signal sur bruit réel doit donc prendre en compte ces distorsions, d'où l'utilisation du paramètre $SINAD$ (*Signal-To-Noise And Distorsion*). Pour comparer le cas idéal avec

le cas réel, on dérive de la formule précédente un calcul de résolution effective appelée *ENOB* (*Effective Number Of Bits*) que l'on compare à la valeur idéale n du convertisseur.

$$ENOB = \frac{SINAD - 1.76}{6.02} \quad (1.6)$$

D'autres paramètres permettent d'évaluer la déviation du composant réel face au composant idéal. La section suivante regroupe ces principaux paramètres.

1.4 Les paramètres de performance du CAN

Les fonctions de transfert réelles des CANs subissent les effets des variations des procédés technologiques utilisés lors de leur conception. Ces variations impliquent un déplacement des tensions de transitions par rapport à leur placement idéal : ce sont les erreurs statiques. D'autres sources d'erreurs impliquent une distorsion du signal analogique : ce sont les erreurs dynamiques. L'ensemble de ces erreurs limitent les performances de la conversion.

1.4.1 Les paramètres statiques

La figure 1.7 illustre les erreurs statiques.

a. **Erreur d'offset**

L'erreur d'offset est un décalage en tension de l'ensemble de la fonction de transfert (figure 1.7(a)). Cette erreur entraîne une erreur constante sur la valeur du code milieu du CAN.

b. **Erreur de gain**

L'erreur de gain correspond à l'écart de pente de la fonction de transfert réelle par rapport à la pente de la fonction de transfert idéale (figure 1.7(b)). Cette erreur change la pleine échelle effective du convertisseur, et donc, la largeur du quantum q .

c. **Erreur de non-linéarité différentielle**

L'erreur de non-linéarité différentielle (NLD) est définie pour chaque code i comme étant la différence relative entre la valeur du quantum réel q_i et le quantum idéal q , le tout normalisé par rapport au quantum q (figure 1.7(c)).

d. **Erreur de non-linéarité intégrale**

L'erreur de non-linéarité intégrale (NLI) du code i correspond au cumul des NLD des codes inférieurs ou égal à i . Cette valeur traduit l'écart de la courbe de conversion par rapport à la droite idéale de conversion ((figure 1.7(d)).

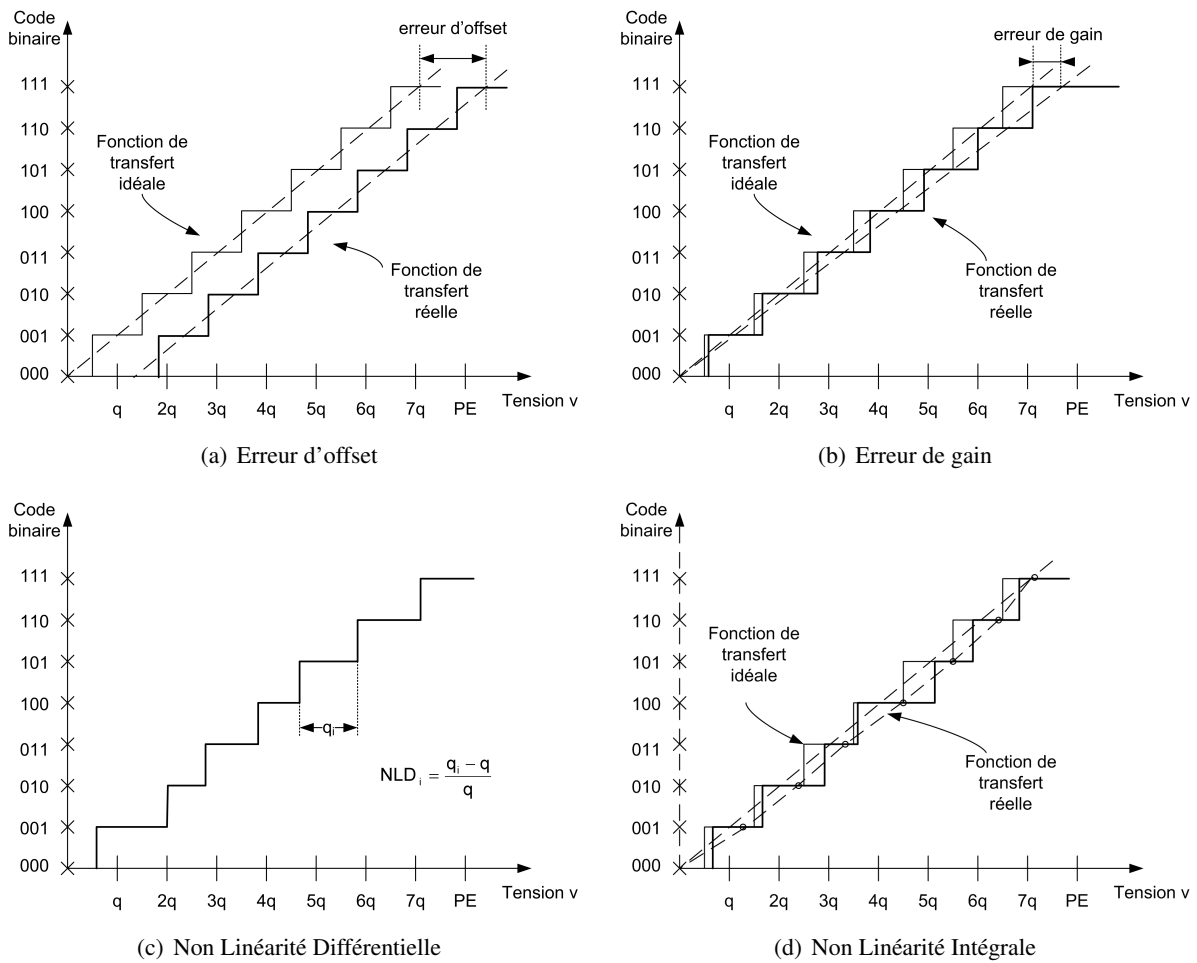


FIG. 1.7 – Erreurs statiques pour un CAN par arrondis

1.4.2 Les paramètres dynamiques

En plus du rapport signal sur bruit précédemment établi, la sensibilité des convertisseurs aux variations brusques du signal analogique se mesure par d'autres paramètres. Nous présenterons ici les deux principaux : le temps d'incertitude à l'ouverture et le taux de distorsion harmonique. Les autres paramètres usuels sont définis en détails dans le standard IEEE 1241 [MotIIMS01].

a. Le temps d'incertitude à l'ouverture

Le temps d'incertitude à l'ouverture, ou *jitter* [SAW89] est lié au fonctionnement de l'échantillonneur-bloqueur. Il s'agit du retard entre le front actif du signal d'échantillonnage et la prise effective de l'échantillon. L'erreur due au temps d'incertitude est directement liée à la fréquence f_{in} du signal analogique. Pour un signal sinusoïdal $s(t) = \frac{PE}{2} \sin(2\pi \cdot f_{in} \cdot t)$, le cas le plus défavorable se situe aux passages à zéro, la dérivée du signal y étant maximale. Le calcul de la pente, $Pente_{max}$, en ce point précis fournit l'expression suivante :

$$Pente_{max} = \pi \cdot f_{in} \cdot PE \quad (1.7)$$

En se référant à la figure 1.3, le temps maximal d'incertitude à l'ouverture τ_{jmax} correspond à la

1.4. LES PARAMÈTRES DE PERFORMANCE DU CAN

valeur de Δt pour laquelle Δv vaut $\frac{q}{2}$. Pour un temps τ_j inférieur à τ_{jmax} , la valeur échantillonnée se trouve obligatoirement dans un intervalle d'erreur de $\pm\frac{q}{2}$, dans le cas contraire, l'effet observé est assimilable à l'ajout d'un bruit sur le signal d'entrée [SAW89]. La valeur τ_{jmax} est donnée par l'expression suivante :

$$\begin{aligned}\tau_{jmax} &= \frac{\frac{q}{2}}{Pente_{max}} \\ \tau_{jmax} &= \frac{\frac{q}{2}}{\pi \cdot f_{in} \cdot PE} \\ \tau_{jmax} &= \frac{1}{2^{n+1} \pi \cdot f_{in}}\end{aligned}\tag{1.8}$$

Comme dans le cas du SNR , l'équation précédente relie le temps d'incertitude à l'ouverture à la résolution n du convertisseur. Une mesure de ce temps permet donc de calculer un équivalent de nombre de bits effectifs, N_{jitter} , que l'on peut comparer aux n bits du convertisseur idéal :

$$N_{jitter} = \log_2 \frac{1}{\pi f_{in} \tau_{jmax}} - 1\tag{1.9}$$

La conception d'E/B respectant la contrainte sur τ_{jmax} est donc primordiale pour l'obtention d'un convertisseur fonctionnel sur une large bande de fréquence de signal analogique d'entrée.

b. Le taux de distorsion harmonique

Le taux de distorsion harmonique (THD) représente le rapport de la puissance des raies harmoniques générées par les non-linéarités du convertisseur, sur la puissance de la raie fondamentale. Les harmoniques sont généralement analysées à travers un traitement numérique sur les codes de sorties. La Transformée de Fourier Discrète permet d'obtenir facilement la puissance de ces raies (figure 1.8). La THD est alors la racine carrée de la somme quadratique des amplitudes des raies harmoniques divisée par l'amplitude du fondamental [MotIIMS01]. Plus ce rapport est faible, plus le CAN possède un comportement linéaire.

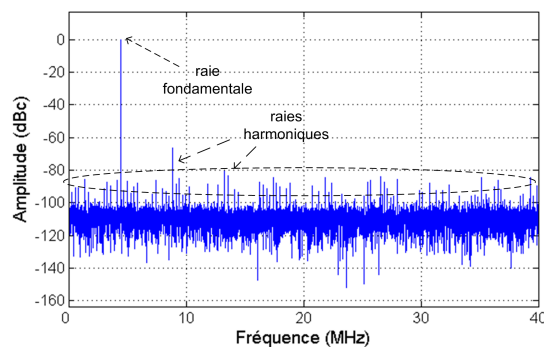


FIG. 1.8 – Spectre de puissance effectué sur les codes de sortie du CAN

1.5 Architectures des Convertisseurs Analogique Numérique

Afin de mieux comprendre les difficultés de conception des CAN, quelques architectures utilisées pour la conversion sont présentées dans cette section. La liste est non exhaustive, mais permet de comprendre l'évolution du convertisseur à partir de l'architecture la plus simple, le *flash*, vers des architectures plus élaborées, comme le *Folding & Interpolation*, qui est l'architecture du véhicule de test de cette étude.

1.5.1 Architecture Flash

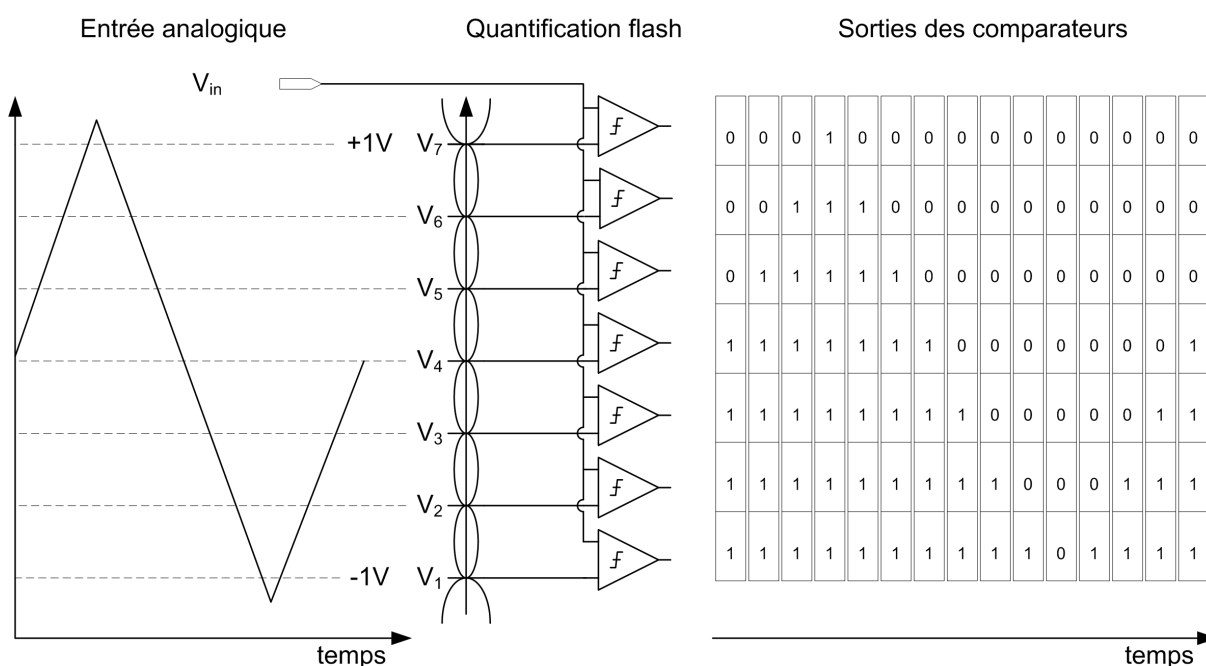


FIG. 1.9 – Quantification flash avec $n = 3$

La technique de conversion la plus simple est celle basée sur l'architecture parallèle dite *flash* [AIW⁺87, GKHK91, ISM⁺84, YANY84]. Un comparateur est associé à chaque tension de transition V_i , ce qui implique autant de comparateurs que de tensions de référence. Ainsi pour un convertisseur de résolution n , la quantification flash utilise $2^n - 1$ comparateurs. La figure 1.9 représente la quantification flash du signal d'entrée V_{in} pour $n = 3$.

Cette technique est souvent identifiée par le nom de *code thermomètre*, car chaque comparateur s'apparente à une graduation présente sur un thermomètre. Quand le mercure atteint une graduation, les graduations inférieures sont également atteintes. Il en va de même dans le cas du convertisseur flash : quand un comparateur est à l'état haut, tous les comparateurs ayant une tension de référence inférieure, le sont également.

Afin de pouvoir utiliser la valeur de sortie d'un comparateur, un second comparateur est nécessaire pour pouvoir déterminer la valeur logique 0 ou 1. Pour remédier aux problèmes de dérives de tensions de référence et d'offset de ces seconds comparateurs, on préfère utiliser une paire différentielle en lieu et place du comparateur de quantification, et ainsi de pouvoir utiliser les sorties différentielles de la paire.

1.5. ARCHITECTURES DES CONVERTISSEURS ANALOGIQUE NUMÉRIQUE

La figure 1.10 illustre le schéma d'une paire différentielle, ainsi qu'une représentation des tensions complémentaires de sortie V_{out_p} et V_{out_n} en fonction de la valeur de V_{in} . Le simple fait d'évaluer la différence $\Delta V = V_{out_p} - V_{out_n}$ permet de connaître précisément la valeur logique de la sortie du comparateur. Si $\Delta V > 0$, alors le comparateur est en saturation positive, tandis que si $\Delta V < 0$, le comparateur est en saturation négative. Le passage par zéro (*zero-crossing*) de ΔV correspond bien à la tension de transition V_i . Les circuits logiques destinés à l'encodage des valeurs de sortie des comparateurs n'ont donc juste qu'à procéder à la différence des tensions de sortie de chaque comparateur et non à une comparaison des niveaux de tensions.

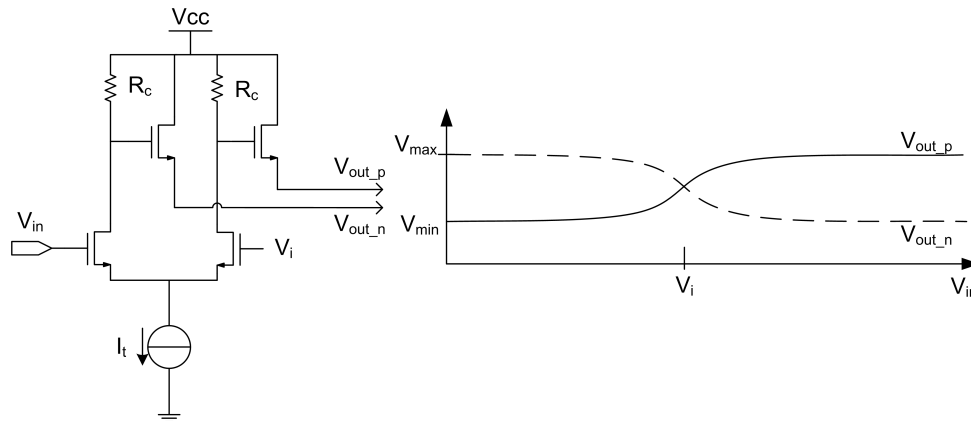


FIG. 1.10 – Principe de fonctionnement d'une paire différentielle

La conversion de type flash est de loin la plus simple de tous les procédés de conversion. Elle ne nécessite, de par son architecture, que d'un seul cycle d'horloge pour générer le code de sortie. Sa simplicité lui permet d'assurer une bonne linéarité de conversion même pour des hautes fréquences d'échantillonnage (jusqu'à 24 GSPS (*Giga Samples Per Secondes*) pour un convertisseur 3bits [NNI⁺04]). Cependant, l'un des principaux inconvénients de cette architecture est la loi exponentielle qui relie le nombre de comparateurs nécessaires à la résolution n du convertisseur. La consommation et la surface requise suivent par conséquent cette même loi exponentielle, ce qui limite ce type d'architecture à une résolution inférieure à 10bits. Des distorsions apparaissent également du fait de la grande capacité de l'étage d'entrée liée à une répartition du signal sur les 2^{n-1} comparateurs nécessaires. La distribution de l'horloge au sein du composant augmente la complexité de conception.

Pour palier à ces effets, de nouvelles architectures ont été développées. Les paragraphes suivants exposent brièvement certaines des ces architectures.

1.5.2 Architecture Flash à Interpolation

Les convertisseurs à interpolation ont pour but de diminuer la capacité d'entrée et le nombre de comparateurs utilisés dans une architecture flash [KMNS93]. La quantification flash à paire différentielle est illustrée sur la figure 1.11 pour trois tensions de transitions successives. Les fonctions de transfert des trois comparateurs sont schématisées par une zone de transition linéaire. L'intersection de la fonction de transfert positive et de la fonction de transfert négative fournit la tension de transition V_i .

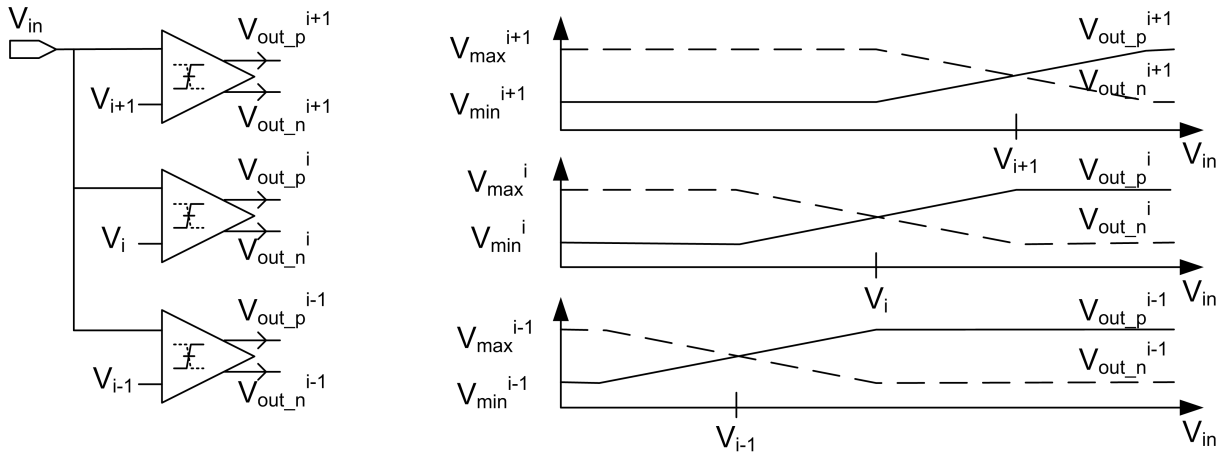


FIG. 1.11 – Quantification flash pour 3 tensions de transition

Le simple fait de substituer le comparateur du milieu par une échelle de résistances R (figure 1.12) permet de recréer le passage à zéro manquant. Les zones linéaires des comparateurs d'indice $i - 1$ et $i + 1$ permettent de recréer linéairement les fonctions de transfert du comparateur d'indice i . Une erreur ΔR sur une des résistances modifiera l'allure de la fonction de transfert interpolée. Cet impact sera exposé dans une section suivante.

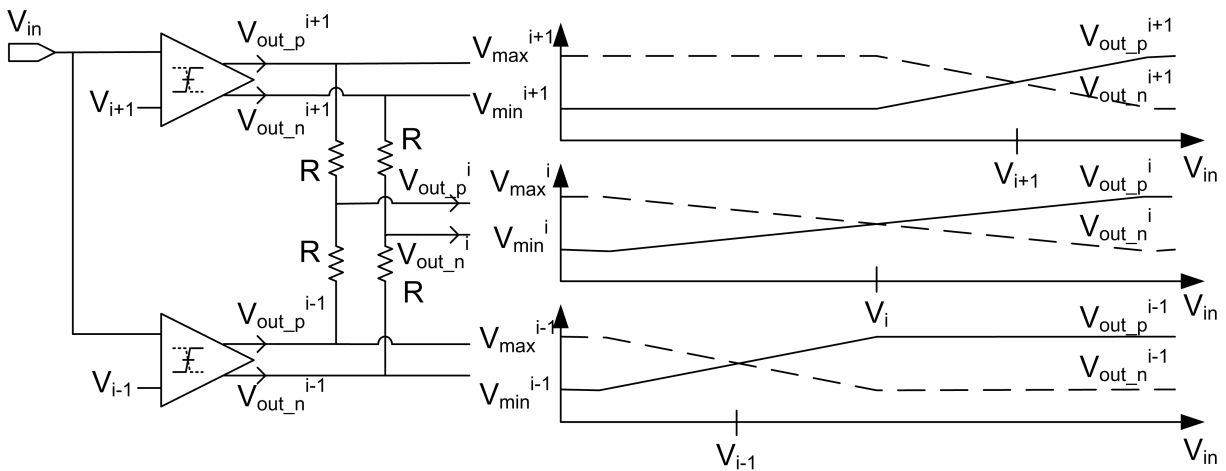


FIG. 1.12 – Quantification avec interpolation résistive

Ce principe d'interpolation peut également s'appliquer pour les codes V_{i-1} et V_{i+1} . De manière générale, les architectures interpolées ne conservent qu'un comparateur sur cinq par rapport à l'architecture flash. Cela revient à réduire fortement la consommation du composant, sa capacité d'entrée et sa complexité de câblage [KMNS93, RS96].

1.5.3 Architecture par partitionnement

Une autre façon de réduire la taille et la consommation du convertisseur tout en conservant sa résolution, consiste à décomposer sa fonction de transfert classique en deux, ou plusieurs sous-fonctions. Ce prin-

1.5. ARCHITECTURES DES CONVERTISSEURS ANALOGIQUE NUMÉRIQUE

cipe est illustré sur la figure 1.13 dans le cadre d'une architecture *pipeline*.

Le convertisseur utilisé dans le cas présent est un convertisseur de résolution $n = 6$ bits, de pas de quantification q et de pleine échelle $PE = 64q$. Sur la figure 1.14(a), la fonction de transfert du convertisseur classique flash est représentée. Pour un souci de clarté, seules les tensions de transitions multiples de $8q$ y sont visibles, ainsi que les codes binaires associés.

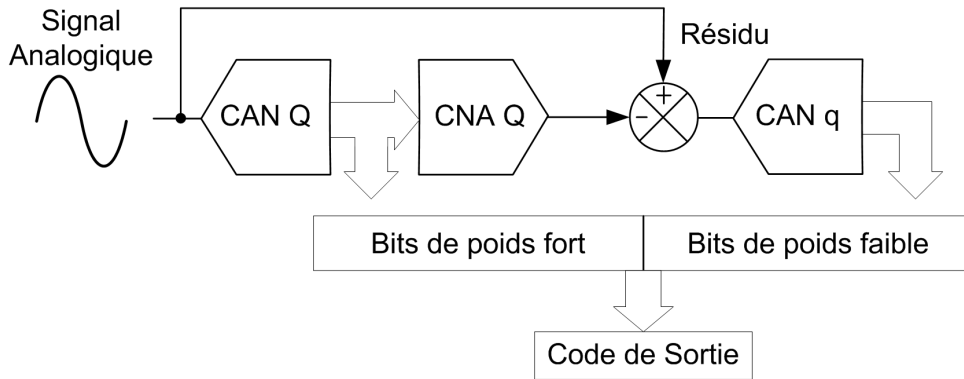


FIG. 1.13 – Principe du convertisseur flash à deux étages

La décomposition de la fonction de transfert représentée sur la figure 1.14(b), se traduit par une conversion en trois étapes (figure 1.13). Premièrement un convertisseur flash de résolution $n = 3$ bits, de quantum $Q = 8q$ et de pleine échelle $PE = 8Q = 64q$ est utilisé pour procéder à une conversion grossière du signal d'entrée. Deuxièmement, cette conversion grossière est retraduite en valeur analogique par un CNA de 3 bits afin d'être soustraite à la valeur analogique d'entrée. Cette soustraction donne lieu à un signal analogique appelé *résidu*, qui possède une amplitude maximale crête-à-crête de $Q = 8q$ Volts. Finalement, ce résidu est envoyé à l'entrée d'un second convertisseur de résolution $n = 3$ bits, de quantum q et de pleine échelle $PE = Q = 8q$. Le fait que ce résidu ait une dynamique constante quelque soit la valeur du signal d'entrée permet de n'utiliser dans cette dernière étape qu'un seul et unique convertisseur. Ce principe de décomposition illustre le fonctionnement des architectures de type *pipeline* [CG95, CA93, SG88].

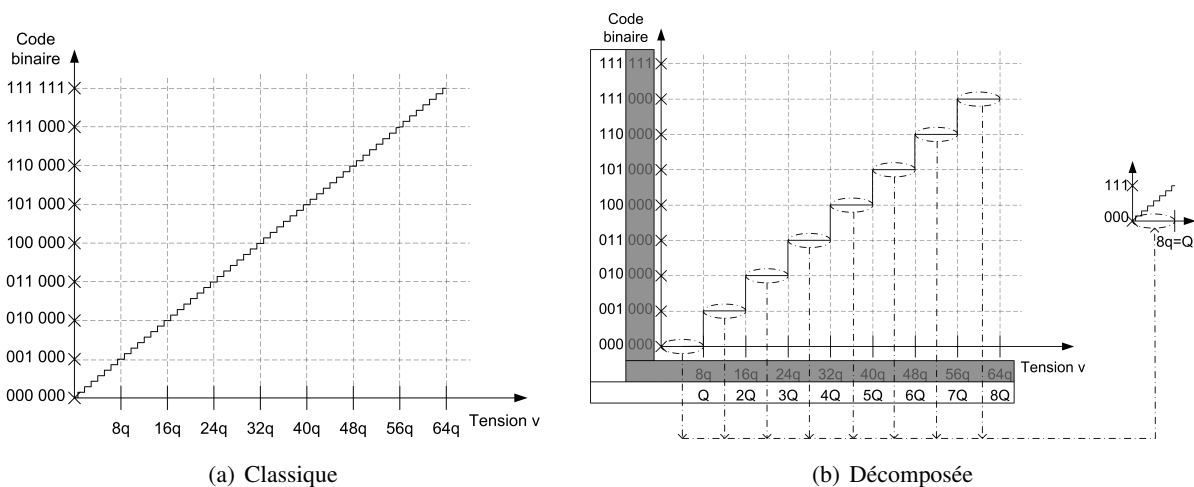


FIG. 1.14 – Fonction de transfert d'un pipeline par troncature

Le tableau 1.1 résume les ressources nécessaires pour la conversion classique de type flash et la conversion de type pipeline, pour un convertisseur de résolution $n = 6$ bits. Les comparateurs sont divisés en deux ensembles, selon la sensibilité requise pour la conversion. En effet, un comparateur de sensibilité $Q = 8q$ est plus facile à reproduire qu'un comparateur de sensibilité q . Réduire le nombre de ces derniers est donc un avantage significatif en terme de reproductibilité lors de la fabrication.

	Flash	Pipeline
Comparateurs de sensibilité q	$2^6 - 1$	$2^3 - 1$
Comparateurs de sensibilité $Q = 8q$	0	$2^3 - 1$
Utilisation d'un CNA	Non	Oui
Latence de conversion	1 cycle d'horloge	> 2 cycles d'horloge

TAB. 1.1 – Ressources nécessaires pour un convertisseur de résolution $n = 6$ bits.

Les inconvénients des architectures pipeline sont l'utilisation d'un CNA et le temps de latence imposé par la double conversion. Pour palier à ces problèmes, d'autres architectures ont été mises au point.

1.5.4 Architecture à Repliection

Le principe du convertisseur à repliection (*folding*) a été introduit pour la première fois par Arbel et Kurz en 1975 [AK75] et a ensuite été amélioré au fur et à mesure de son utilisation, notamment par van de Plassche [vdP03] et Vorenkamp [VR97]. Ce convertisseur opère de manière similaire au convertisseur pipeline de la figure 1.13, à la différence près que le résidu est généré par un bloc de repliection analogique comme indiqué sur la figure 1.15. L'utilisation de ce bloc évite l'usage du CNA et la soustraction des signaux. Un autre intérêt majeur de cette architecture provient de la possibilité de faire fonctionner tous les éléments en parallèle et donc de réduire la latence à un cycle d'horloge. Les paragraphes suivants exposent le fonctionnement du bloc de repliection.

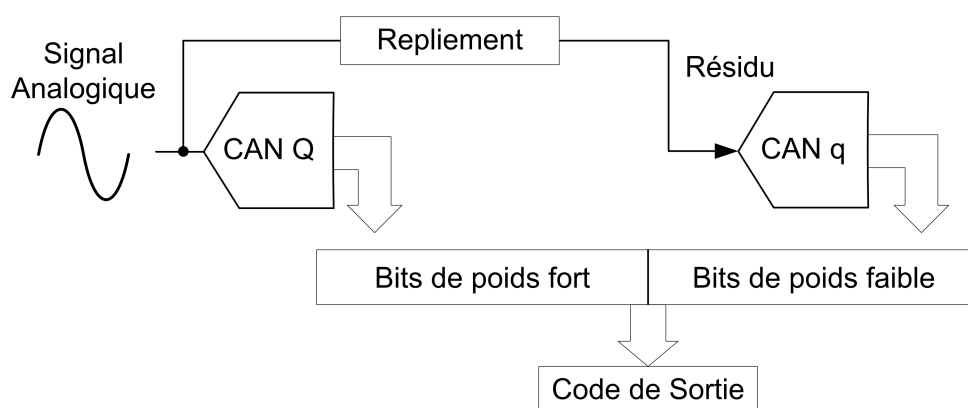


FIG. 1.15 – Principe du Convertisseur à Repliection

Amplificateur à Repliement Simple

Le rôle de l'amplificateur à repliement est de fournir le signal résidu. Dans le cas idéal, ce bloc analogique fournit une fonction de transfert "en dents de scie" comme indiqué sur la figure 1.16(a). Cette implémentation est difficilement réalisable du fait de la présence des discontinuités lors du changement de code des bits de poids fort. La forme triangulaire décrite sur la figure 1.16(b) lui est préférée. Son implémentation est possible à l'aide de montages réalisés à partir de diodes [Gil82, PG79] ou de miroirs de courants [LSS00].

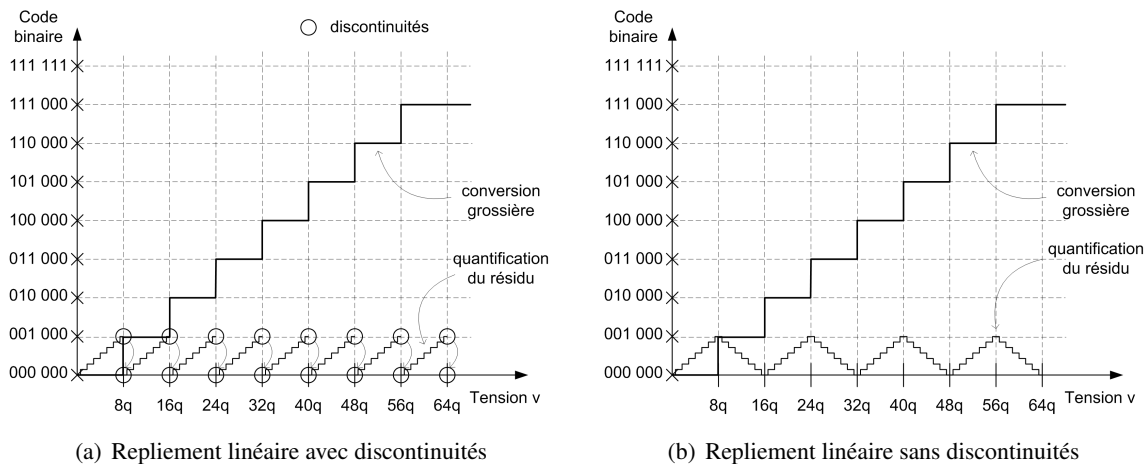


FIG. 1.16 – Fonction de transfert du Convertisseur à Repliement

L'assemblage alterné de paires différentielles (figure 1.10) permet d'obtenir une forme pseudo-triangulaire. La figure 1.17 représente l'agencement alterné de trois paires différentielles. Les lignes de sorties sont tout simplement reliées entre elles de façon inversée l'une par rapport à l'autre.

Cependant, l'obtention de fonctions de transfert totalement linéaires à partir d'éléments analogiques s'avère délicate à mettre en oeuvre. Dans le cas de nos paires différentielles la zone linéaire est réduite à l'intervalle de changement d'état autour de la tension de référence V_i (figure 1.18) correspondant aux tensions de transition du CAN Q. Au delà de cette zone linéaire, la fonction de transfert s'arrondit, transformant l'allure triangulaire en une allure pseudo-sinusoïdale.

Pour palier à ce problème, la technique du double repliement est habituellement utilisée.

Amplificateur à Double Repliement

Un repliement idéal requiert une fonction de transfert triangulaire parfaite pour procéder à une conversion performante. La première courbe de la figure 1.19 illustre ce cas idéal, la conversion du signal résiduel s'effectuant dans notre exemple par un CAN ayant 3 bits de résolution.

Comme indiqué précédemment, les différents montages analogiques permettent uniquement d'obtenir des fonctions de transfert partiellement linéaires. La seconde courbe de la figure 1.19 illustre la conversion liée à un repliement simple. La zone hachurée indique la plage linéaire de la fonction de transfert. Il apparaît clairement sur cette figure que la conversion ne peut être correcte que sur les zones $a1$, $b1$ et $c1$,

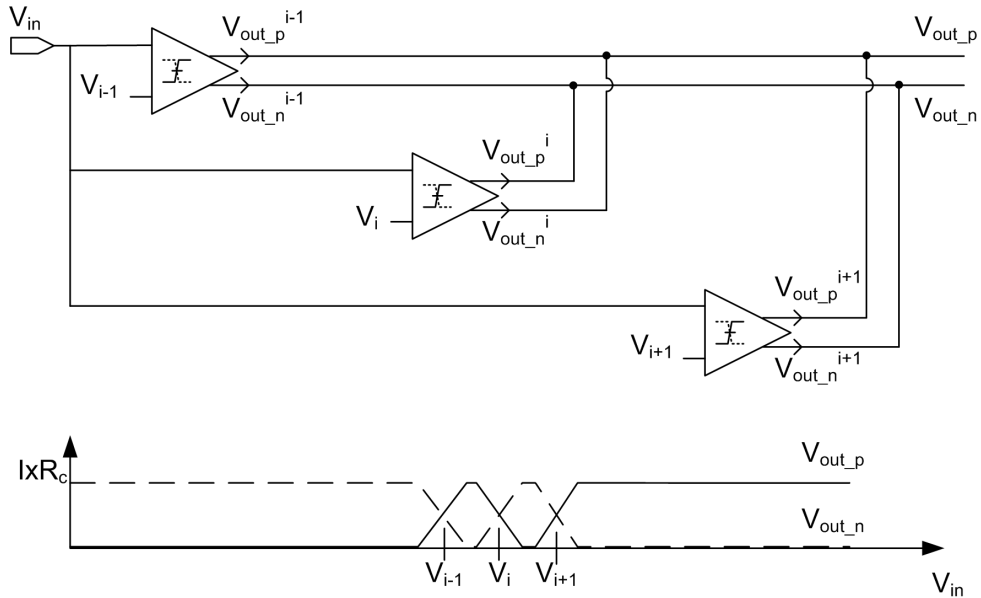


FIG. 1.17 – Montage de paires différentielles

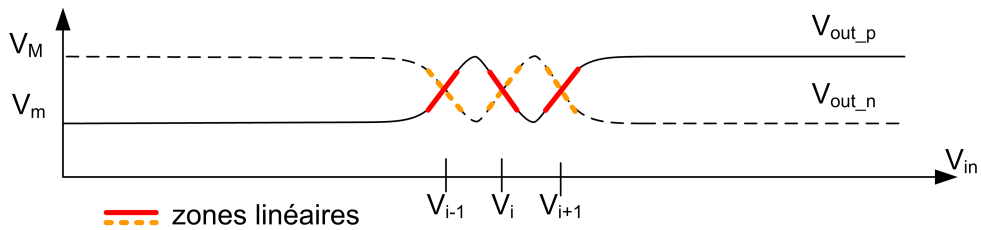


FIG. 1.18 – Non linéarité du montage différentiel

alors que l'on désire convertir les zones A , B et C .

L'idée du double repliement est d'utiliser les zones linéaires d'une seconde fonction de transfert, dont les tensions de référence V'_i ont été légèrement décalées par rapport aux tensions V_i du premier bloc de repliement. La troisième courbe de la figure 1.19 illustre ce principe. Le second repliement permet de couvrir les zones de conversion a_2 , b_2 et c_2 . Ainsi, l'assemblage de la conversion sur 2×2 bits des couples de zones (a_1, a_2) , (b_1, b_2) et (c_1, c_2) permet de retrouver la conversion sur 3 bits des zones A , B et C .

Par récursivité, les zones originelles A , B et C peuvent être décomposées en 8 sous-zones que l'on nommera $(a_0, \dots, a_7, b_0, \dots, b_7, c_0, \dots, c_7)$. L'utilisation de 8 fonctions de repliement (figure 1.20) permet d'associer à chaque zone linéaire la conversion d'un simple et unique bit $(a_0, \dots, a_7, b_0, \dots, b_7, c_0, \dots, c_7)$, dégénéralisant ainsi la conversion en aval du signal en un simple comparateur de niveaux. De plus, comme les signaux issus des blocs de repliement sont de nature différentielle, ces comparateurs fonctionnent selon le même principe que celui exposé dans l'architecture flash, à savoir en analysant le passage à zéro (cf §1.5.1).

La figure 1.21 représente les seules zones (A, B) de la fonction de transfert principale, ainsi que les 8

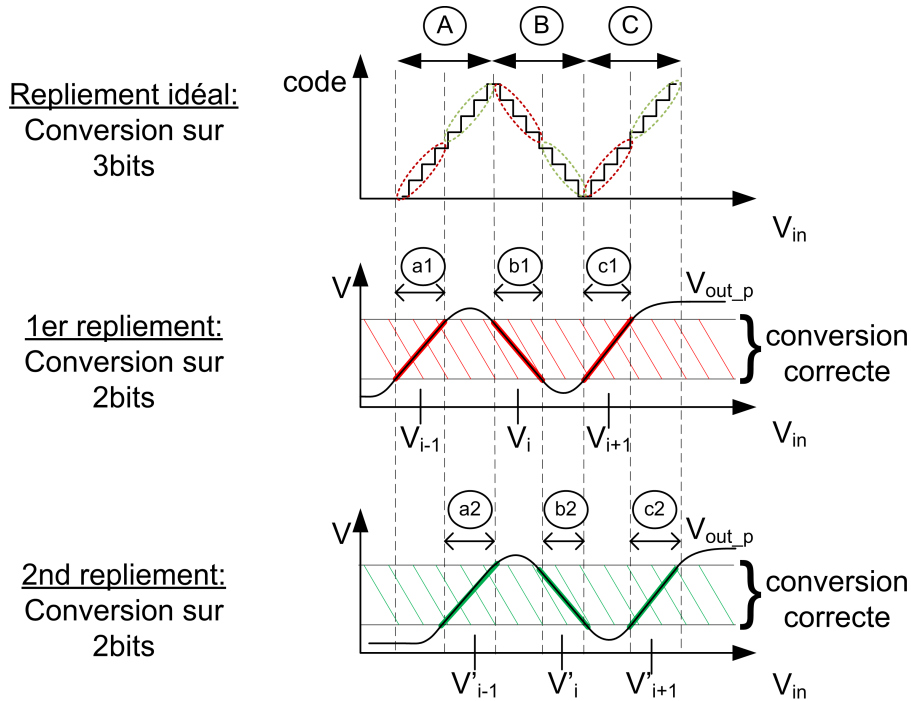


FIG. 1.19 – Double repliement

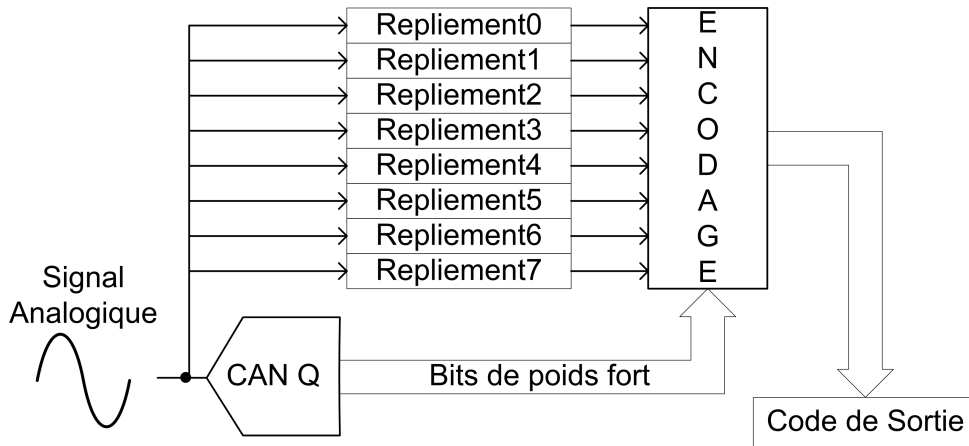


FIG. 1.20 – Architecture avec multiples repliements

fonctions de repliements positives et négatives issues des blocs de repliements.

Chaque passage à zéro correspond à un changement de bit dans la fonction de transfert. Pour chaque signal replié, un seul comparateur est nécessaire. Ce comparateur passe en saturation positive lorsque le signal positif est supérieur au signal négatif et en saturation négative pour le cas inverse. Pour notre exemple, 8 comparateurs indiquent à chaque instant la valeur de la différence *fonction positive - fonction négative*. La figure 1.22 représente l'état des comparateurs associés aux fonctions de repliements nommées ($R.0 \dots R.7$), en fonction de l'amplitude du signal d'entrée V_{in} . Un simple encodage de ces valeurs permet, à l'aide des bits de poids fort, de construire le code numérique de sortie.

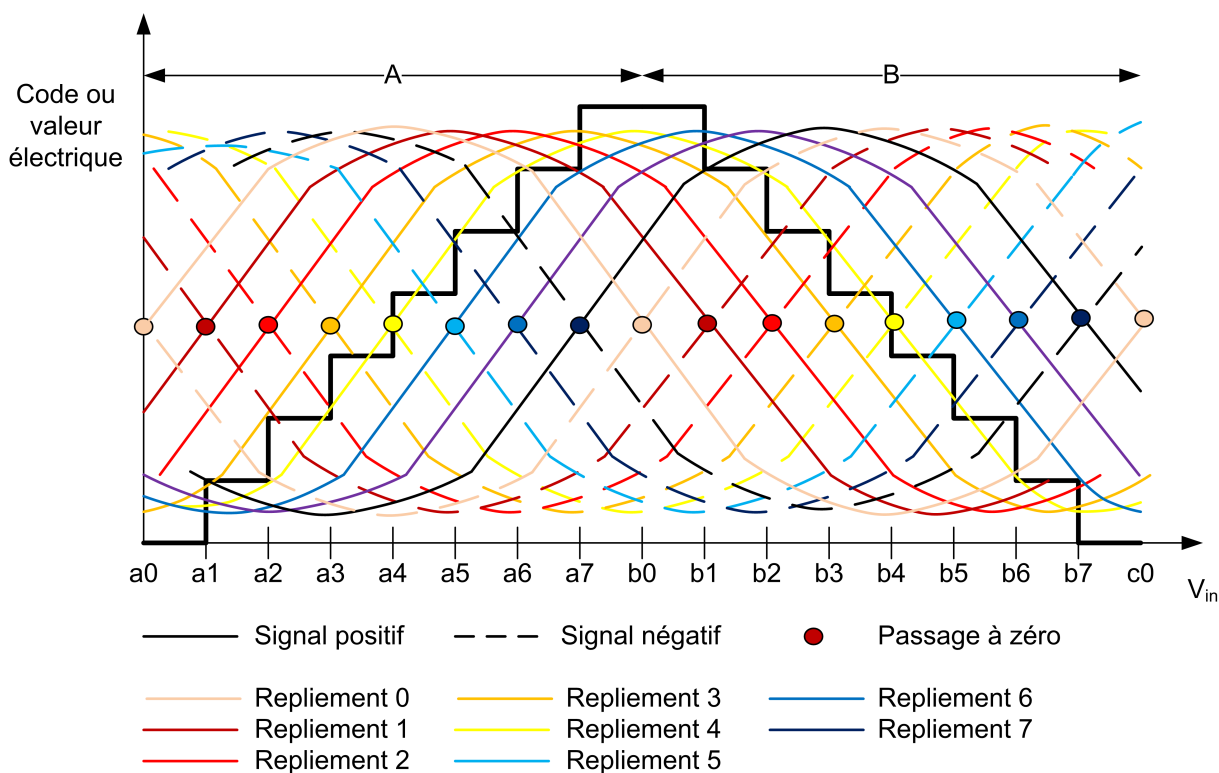


FIG. 1.21 – Multiples repliements

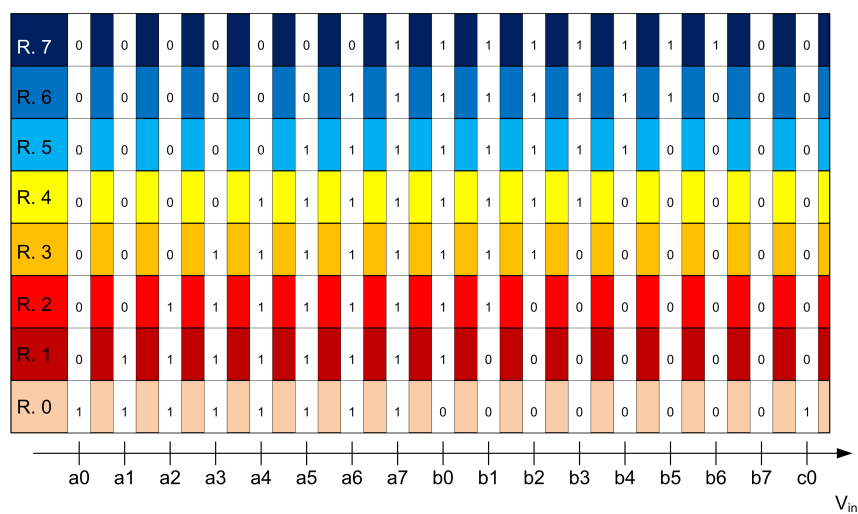


FIG. 1.22 – Valeur des comparateurs associés aux passages à zéro des signaux repliés

Interpolation des fonctions de repliement

De la même façon que pour l'architecture Flash interpolée, une interpolation résistive peut être utilisée afin de recréer les fonctions de repliement [vdPB88]. La figure 1.23 illustre l'interpolation entre la fonction positive du second amplificateur de repliement ($R.2^+$) et la fonction positive du quatrième amplificateur de repliement ($R.4^+$). La même interpolation est utilisée pour les fonctions négatives afin de

retrouver le passage à zéro nécessaire à la conversion.

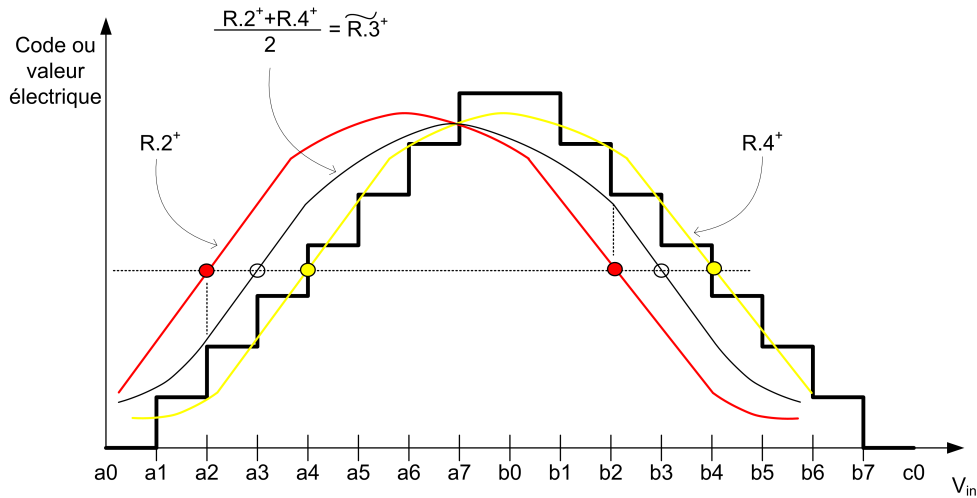


FIG. 1.23 – Interpolation entre deux signaux repliés

La nouvelle architecture du convertisseur à repliement interpolé se présente désormais comme indiqué par la figure 1.24. Trois étages de repliements ont été supprimés et remplacés par de simples résistances. Cette substitution permet de réduire la consommation globale du convertisseur.

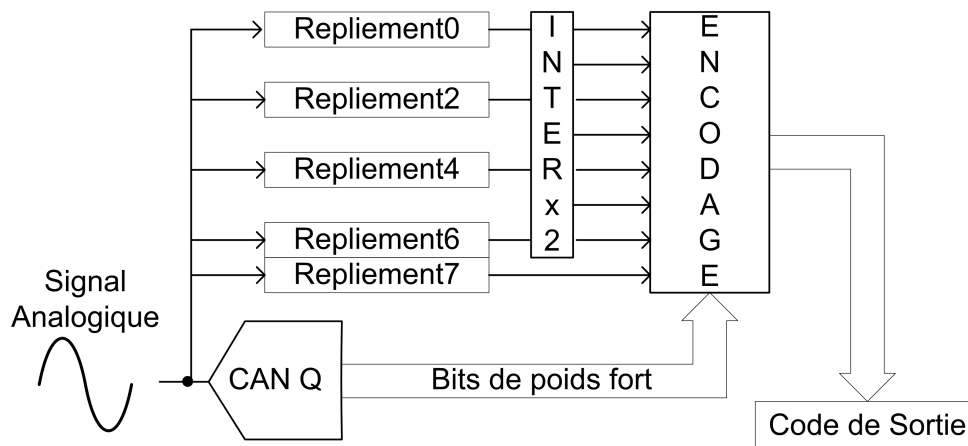


FIG. 1.24 – Architecture à Repliement interpolé

Les techniques classique d'interpolation génèrent 4 fonctions interpolées entre deux blocs de repliements afin de réduire le nombre de composants nécessaires, et donc la consommation totale du convertisseur [VvdP96]. De nombreuses architectures mettent également en oeuvre un second étage de repliement afin de réduire encore plus cette consommation [FS98, TSE98].

1.5.5 Limitations de l'architecture à repliement et à interpolation

La conception de l'architecture à repliement impose des contraintes fortes sur les différents blocs afin d'obtenir une bonne linéarité pour la conversion. Les paragraphes suivants exposent succinctement quelques-unes de ces contraintes.

Échelle de résistances et offset

Une échelle de résistances est habituellement utilisée pour générer les tensions de références nécessaires aux repliements. La figure 1.25 représente la génération des tensions de références V_{ai} , V_{bi} et V_{ci} pour l'amplificateur de repliement $R.i$. Cette génération s'effectue à l'aide de ponts diviseurs de tensions. Une variation ΔR dans la valeur des résistances, ou l'apparition d'une tension d'offset V_{offset} dans la paire différentielle implique un décalage des signaux de sorties comme indiqué sur la figure 1.26.

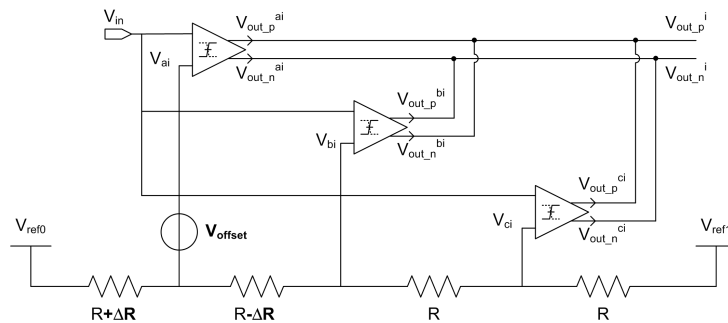


FIG. 1.25 – Échelle de résistances et erreurs associées à l'étage d'entrée

Ce décalage modifie la position du passage à zéro et donc modifie la valeur idéale du pas de quantification. Pour que l'influence de cet offset soit minimale, sa valeur doit être inférieure au pas de quantification. Les relations suivantes doivent donc être vérifiées :

$$V_{offset} < \frac{PE}{2^n} \quad (1.10)$$

$$\frac{\Delta R}{R} < \frac{1}{2^n} \quad (1.11)$$

Gain

La construction des paires différentielles, utilise un assemblage de transistors dont les valeurs de gain influencent la pente du signal de sortie. Pour des transistors idéaux, la pente de la caractéristique de transfert du repliement dans leur zone linéaire est connue. La variation de la pente n'interfère pas sur la position du passage à zéro. Par contre, lors de l'interpolation, une erreur de gain sur une caractéristique servant à interpoler, impacte la position du passage à zéro de la fonction interpolée. La figure 1.27 illustre cet impact.

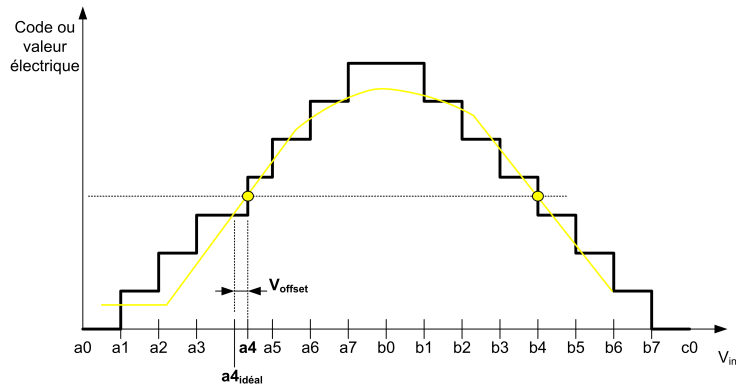


FIG. 1.26 – Influence des erreurs

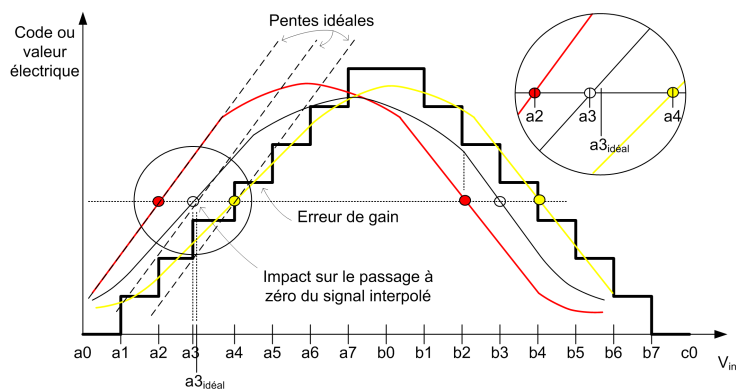


FIG. 1.27 – Impact d'une erreur de gain

Offset des comparateurs

Les signaux issus des étages de repliement (et des interpolations) sont envoyés à l'entrée de comparateurs chargés d'évaluer la différence ΔV . Si cette différence est positive (négative), le comparateur sature positivement (négativement). Un offset ϵ sur l'une des voies entraîne un décalage δ sur la valeur de décision, ce qui a pour effet de fausser la conversion. La figure 1.28 illustre cette erreur.

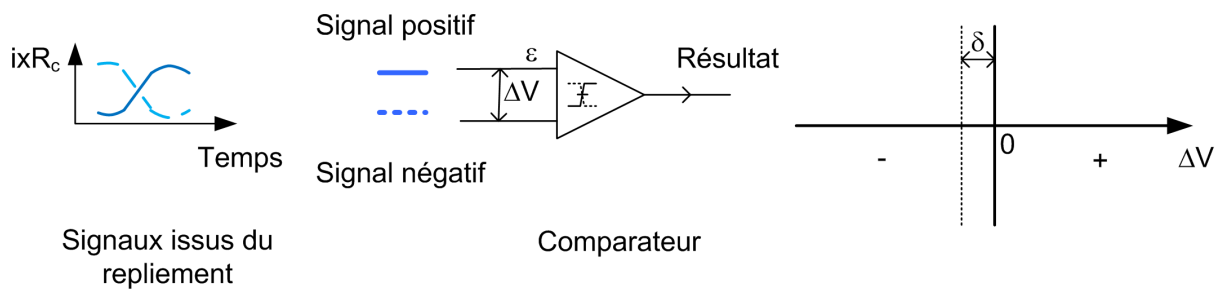


FIG. 1.28 – Erreur sur les comparateurs

Problématique linéarité/vitesse

Le transistor permet de réaliser de nombreuses fonctions élémentaires au sein de l'architecture de conversion. Son utilisation pour les paires différentielles permet de réaliser les fonctions de repliement du signal analogique. La figure 1.29 illustre la structure d'un transistor MOS. L'utilisation la plus contraignante, d'un point de vue conception, est l'utilisation en amplificateur. Pour assurer une amplification du signal linéaire, le concepteur doit s'assurer que l'effet des différents bruits (thermique, 1/f, issu du substrat,...) soit négligeable. Nous ne rentrerons pas ici dans le détail de la conception d'un transistor MOS, mais nous accepterons le fait que les paramètres critiques du transistor tels la tension de transition ($V_{threshold}$) et le facteur de courant β sont liés à la géométrie du transistor par un facteur \sqrt{WL} , où L représente la longueur du canal et W sa largeur [PDW89].

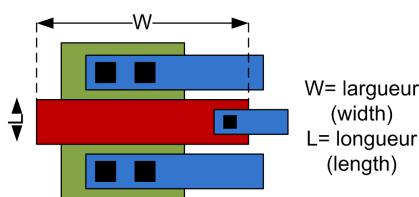


FIG. 1.29 – Transistor MOS

La figure 1.30 met en évidence la variance d'un des paramètres précédents (que l'on notera P) en fonction de la géométrie : plus l'aire WL du transistor est grande, plus la variance $\sigma_{\Delta P}$ sera faible, et moins le transistor sera sensible aux divers bruits. Cependant, plus la surface est grande, plus le temps d'établissement du transistor sera long. Le concepteur doit donc faire un choix entre vitesse et linéarité.

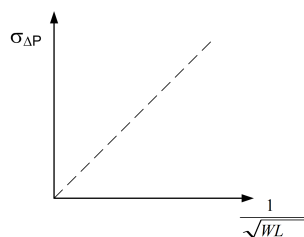


FIG. 1.30 – Variance typique en fonction de la géométrie

1.6 Nouvelle stratégie de conception et conclusion

Tout au long des paragraphes précédents, nous avons établi que les variations de procédés et l'augmentation de la vitesse de fonctionnement entraînent l'apparition de non-linéarités au sein du convertisseur. Pour remédier à ces problèmes, les concepteurs doivent augmenter la taille des divers composants sensibles, afin de les rendre robustes aux variations. L'inconvénient de cette technique est l'augmentation du temps d'établissement au sein des différents blocs de l'architecture, ce qui impose une diminution

1.6. NOUVELLE STRATÉGIE DE CONCEPTION ET CONCLUSION

intrinsèque de la vitesse de fonctionnement du composant. Cette problématique linéarité/vitesse est le principal facteur limitatif d'obtention de convertisseur haute résolution à haute vitesse de fonctionnement. Une meilleure compréhension et une amélioration des procédés de fabrication peuvent permettre de réduire leurs propres variations et donc de repousser un peu plus loin les limites atteignables, mais cette opération est longue, plus coûteuse, et difficile à mettre en oeuvre dans le cadre d'une industrialisation à grande échelle.

Dans le cadre de cette étude, nous considérons la conception de convertisseur d'une manière différente. Au lieu de chercher à optimiser la linéarité du convertisseur et par conséquent, de réduire la vitesse de fonctionnement, le concepteur est autorisé à réduire la taille des différents composants afin d'augmenter la vitesse. Cette réduction introduit alors des nouvelles non-linéarités que l'on tente de compenser par la suite. La figure 1.31(a) schématise un convertisseur conçu avec les paramètres sensibles W , L et R de grande taille, ce qui implique un composant linéaire mais lent. La figure 1.31(b) schématise un convertisseur conçu avec les paramètres sensibles de taille plus réduite, impliquant un composant moins linéaire mais plus rapide. Dans cette conception, une phase de compensation est rajoutée afin de retrouver la linéarité du composant. C'est la phase de compensation qui fait l'objet de la présente thèse.

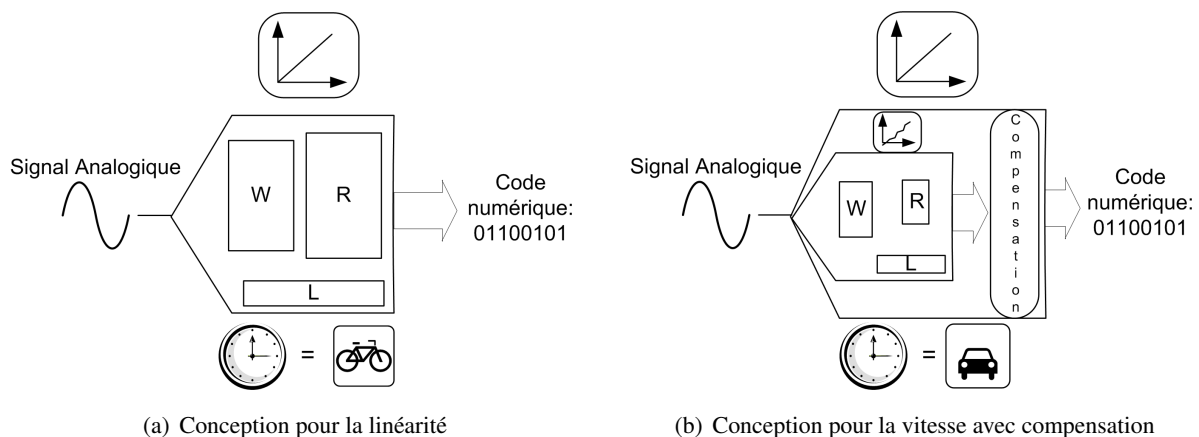


FIG. 1.31 – Différences de conception

Ce processus de compensation peut être implémenté de deux façons différentes :

- Les phénomènes responsables des non-linéarités sont mesurés lors de la fabrication du composant et leurs valeurs stockées au sein du composant. A partir de ces valeurs, une méthode de compensation permet de linéariser le produit. Cette méthode se nomme compensation *offline* car les valeurs des paramètres sont mesurées de façon externe au convertisseur, et elles sont figées une fois pour toute (figure 1.32(a)).
- Les phénomènes responsables des non-linéarités sont mesurées tout au long de la vie du produit, et les valeurs sont donc mises à jour régulièrement. A partir de ces valeurs, une méthode de compensation permet de linéariser le composant. Cette méthode se nomme compensation *online* car les valeurs des paramètres sont mesurées de façon interne au convertisseur (figure 1.32(b)) et permet de la modifier durant la vie du circuit.

Il apparaît au travers de ces illustrations que les méthodes de compensation sont identiques dans les deux options, seule la façon de mesurer les non-linéarités change. Dans le chapitre suivant, nous exposons les

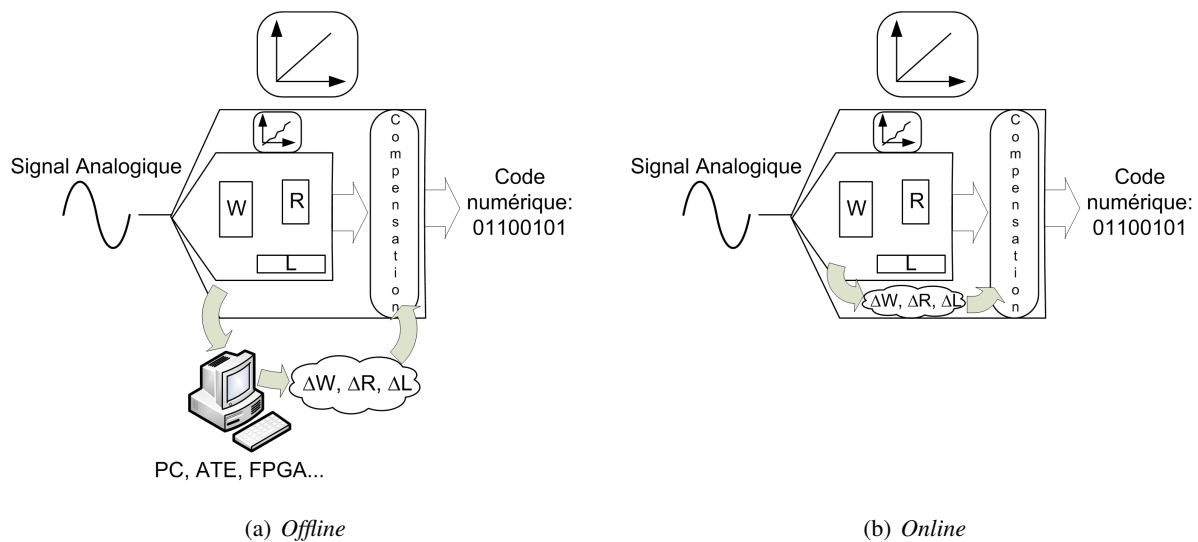


FIG. 1.32 – Méthodes de compensation

différentes méthodes disponibles pour effectuer la compensation.

Chapitre 2

Méthodes de Compensation

Le second chapitre est consacré aux méthodes de compensation couramment présentées dans la littérature. Trois grands types de méthodes sont mis en avant : la correction par table de correspondance, la correction par inversion de modèle, et la correction liée à l'architecture du convertisseur. Après avoir exposé leurs différents principes, leurs avantages et inconvénients sont mis en avant, et une méthode est sélectionnée pour être utilisée dans le contexte de notre étude.

2.1 Introduction

Le chapitre précédent a exposé le besoin de processus de compensation pour l'obtention de convertisseurs haute résolution et à haute vitesse de fonctionnement. Le présent chapitre propose un état de l'art de ces techniques [BDR05].

Dans un premier temps les méthodes basées sur l'utilisation d'une table de correspondance sont exposées. Dans un second temps, nous nous intéresserons aux méthodes basées sur une modélisation des erreurs [ADR04]. Ces deux premières méthodes considèrent le convertisseur comme une "boîte noire", elles traitent les données issues du CAN indépendamment de son architecture. Enfin, la troisième partie traite des méthodes intimement liées à l'architecture du convertisseur à compenser. Dans ce contexte, nous présentons ensuite l'approche choisie pour la compensation des convertisseurs de type à repliement et interpolation.

2.2 Méthodes basées sur l'utilisation d'une table de correspondance

De façon générale, nous utiliserons le terme anglais Look-Up Table (LUT) en lieu et place de table de correspondance. Le terme Look-Up Table trouve son origine dans le monde de l'informatique où son utilisation sert à remplacer un calcul complexe en une simple consultation de valeur. Par exemple, le calcul d'une fonction trigonométrique (ex : sinus) est une procédure lourde qui peut ralentir l'exécution d'un programme. Pour éviter ce ralentissement, une table de correspondance est stockée dans un tableau que le programme vient consulter selon ses besoins. Le programme charge alors la valeur disponible la

plus proche.

Dans le cadre de la correction d'un convertisseur, la façon la plus intuitive de procéder est de substituer le code erroné par sa valeur idéale (figure 2.1(a)). Cette substitution s'effectue par le biais d'une LUT où le code de sortie du convertisseur sert d'adresse, et la valeur stockée dans la table fournit le code de sortie corrigé. Cette méthode se nomme correction statique, car la correction n'utilise qu'une seule information, le code de sortie à l'instant t . Afin d'améliorer cette correction, il faut considérer d'autres éléments susceptibles de rajouter de l'information pour choisir la bonne valeur corrective. Par exemple, la valeur du code de sortie à l'instant $t - 1$ permet de préciser la fréquence du signal d'entrée. La correction est alors dite dynamique et la table de correction devient multidimensionnelle. Les paragraphes suivants rentrent dans les détails des différences entre les deux méthodes.

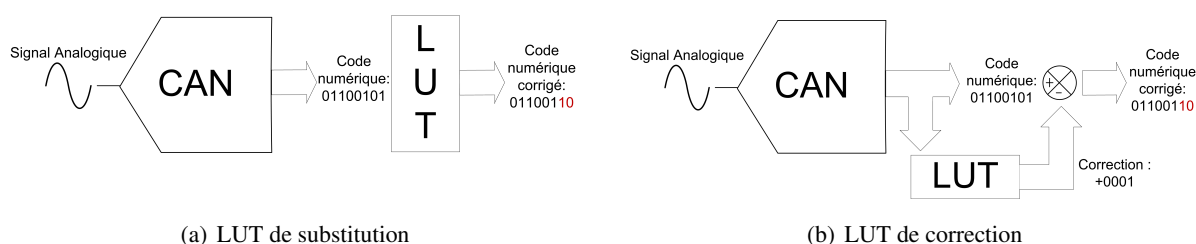


FIG. 2.1 – Schéma d'utilisation de LUT

2.2.1 Correction statique

Dans le cadre d'une correction statique, seule la valeur de l'échantillon de sortie est prise en compte dans la sélection de la valeur corrective à appliquer. La LUT utilise donc un adressage codé sur n bits (n étant la résolution du CAN), et un codage des valeurs sur n bits également. Afin de réduire la taille de la LUT, seuls les termes correctifs peuvent être stockés et ensuite soustraits (figure 2.1(b)). L'adressage de la LUT est alors toujours de n bits, mais le stockage s'effectue à l'aide d'un bit de signe et de 3 ou 4 bits pour les valeurs correctives.

Cette méthode principalement introduite par Irons et Hummels [IHK91] ne fonctionne essentiellement que pour des non-linéarités dites statiques. Un des critères d'évaluation de la correction, est l'amélioration du paramètre *SFDR* (*Spurious Free Dynamic Range*). Ce paramètre est obtenu à partir du spectre du signal échantillonné et traduit la hauteur minimale entre la valeur de la fondamentale et la plus haute des raies parasites ou harmoniques du spectre. Etant donné que les non-linéarités se traduisent spectralement par la présence de raies harmoniques de niveaux élevés, la compensation de ces non-linéarités atténue leurs valeurs. Les résultats présentés dans [HSP00] montrent une amélioration globale de plus de 15dB pour le paramètre *SFDR*, mais cette amélioration est dépendante de la fréquence d'entrée. La technique ne prend pas en compte la variation de comportement des composants électroniques en fonction des fréquences d'entrée et d'échantillonnage. Dans [LASH], les auteurs utilisent un procédé similaire à la Look Up Table à une dimension et mettent en évidence que la correction calculée pour une fréquence donnée continue de fonctionner autour de cette fréquence, mais que son efficacité diminue au fur et à mesure que l'on s'en éloigne. La figure 2.2 illustre la correction d'un convertisseur 10bits échantillonné à 20MSPS, et dont la table de correction a été calculée à la fréquence d'entrée de 2.5MHz. L'amélioration du *SFDR*

est notable lorsque la fréquence d'entrée est voisine de la fréquence de calcul de correction, mais elle devient faible voire négative lorsque l'on s'en éloigne.

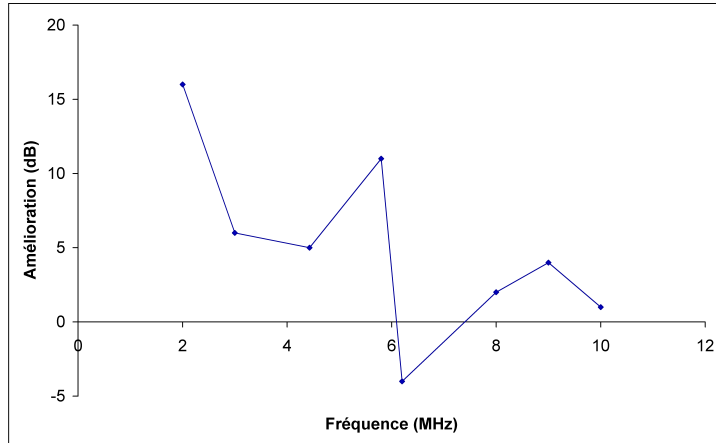


FIG. 2.2 – Amélioration du SFDR en fonction de la fréquence d'entrée.

L'utilisation de ce procédé se limite donc aux convertisseurs dont les non-linéarités sont peu sensibles, voire indépendantes, aux variations de fréquences. Afin de supprimer cette limitation, les techniques de correction dynamique dites de Look-Up Table à plusieurs dimensions ont été développées.

2.2.2 Correction dynamique

Dans le chapitre précédent, les erreurs liées à l'architecture à repliement et interpolation ont été présentées. Si les erreurs d'offset sont, en général, indépendantes de la fréquence d'utilisation du convertisseur, il n'en est pas de même pour les erreurs de gains et d'échantillonnage pour lesquelles le temps de réaction des composants est primordial. Dans [HSP00], les auteurs mettent en évidence la nature dynamique des non-linéarités du convertisseur. Ainsi, lorsque la fréquence de travail ou la fréquence d'entrée du convertisseur augmentent, une baisse significative des performances est observée. Dans le cas de l'échantillonneur/bloqueur (E/B), une fréquence d'entrée élevée implique une vitesse de poursuite lors de la phase d'échantillonnage très élevée. Les figures 2.3 et 2.4 illustrent le comportement de l'E/B en fonction de la fréquence d'entrée, le trait épais représentant le signal d'entrée, et le trait pointillé le signal de sortie de l'E/B. Pour une fréquence faible (fig. 2.3), l'erreur créée sur les échantillons d'indice j et $j - 1$ est faible. Ces erreurs se mesurent par la différence entre les valeurs estimées $\hat{x}[j]$ et $\hat{x}[j - 1]$ et leur valeur idéale $x[j]$ et $x[j - 1]$. Dans le cas d'un signal d'entrée à fréquence plus élevée (fig. 2.4), l'échantillonneur s'avère trop lent pour suivre le signal, et les valeurs estimées sont plus éloignées des valeurs idéales.

La prise en compte de la dynamique du signal d'entrée a donné lieu à de nouvelles techniques d'utilisation des LUT. Deux techniques se différencient : la correction dite d'Espace d'Etat (*State-Space*) et la correction dite de Plan de Phase (*Phase-Plane*). Chacune utilise une ou plusieurs informations supplé-

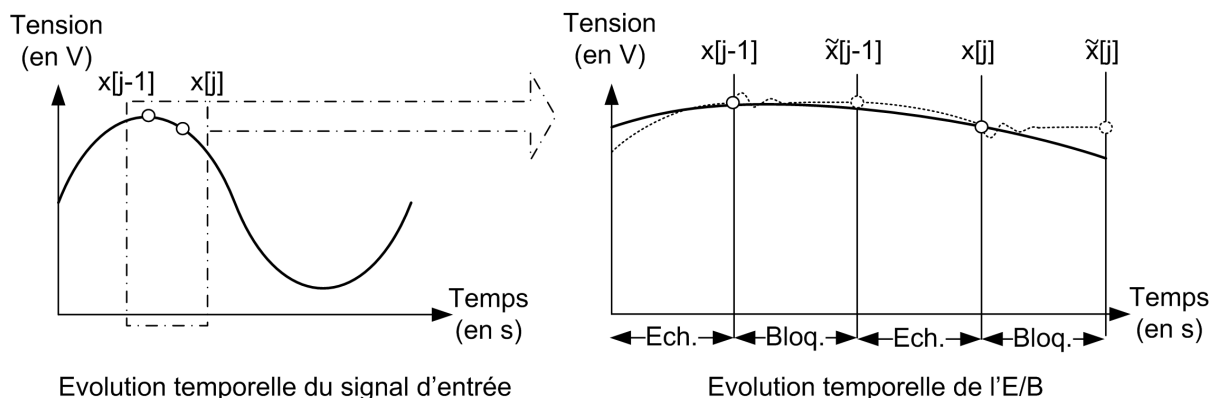


FIG. 2.3 – Erreur de l'E/B pour une fréquence d'entrée faible

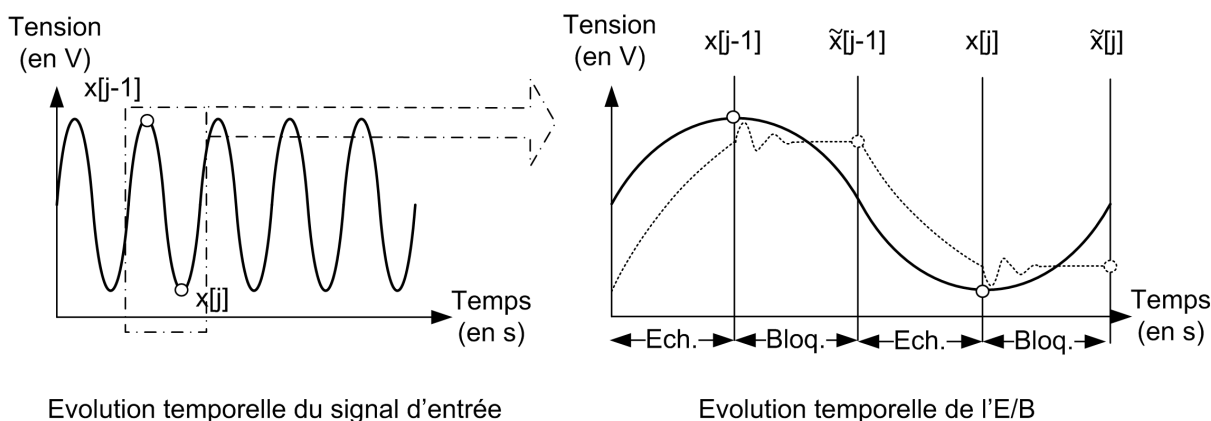


FIG. 2.4 – Erreur de l'E/B pour une fréquence d'entrée élevée

mentaires afin de choisir la valeur de correction en fonction des caractéristiques du signal d'entrée. Les paragraphes suivant exposent les principes de ces deux techniques.

Post-correction d'Espace d'Etat

L'une des façons intuitives de prendre en compte les brusques variations dans le signal d'entrée est de considérer non plus le seul échantillon $x[j]$, mais également l'échantillon précédent $x[j - 1]$ [IHK91], ou de façon plus générale, les K échantillons précédents [Tsi95]. Mathématiquement, l'ensemble des $K + 1$ échantillons constituent un espace de dimension $K + 1$, de plus, l'information supplémentaire apportée par les K échantillons représente l'état du système, ce qui justifie l'appellation Espace d'Etat de cette technique. La figure 2.5 illustre le cas $K = 1$. L'erreur est mesurée pour chaque couple $(x[j], x[j - 1])$ et forme ainsi une surface en trois dimensions. Le calcul de ces erreurs fait intervenir de nombreux stimuli permettant de couvrir l'ensemble des couples $(x[j], x[j - 1])$ susceptibles d'apparaître lors d'un comportement applicatif normal. La correction s'effectue alors en soustrayant au code $x[j]$ la valeur de l'erreur fournie par les coordonnées $(x[j], x[j - 1])$.

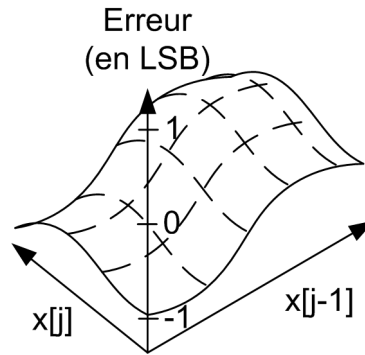


FIG. 2.5 – Espace d’Etat à deux dimensions (K=1)

De façon concrète, l’adressage de la LUT s’effectue à partir de la concaténation des $K + 1$ échantillons [TMBSL02]. Dans le cadre d’un convertisseur de résolution n , le codage basique de l’adresse s’effectue alors sur $B = (K + 1) \times n$ bits. En plus de consommer beaucoup de mémoire, les temps d’accès à la table de stockage deviennent également un problème. En effet, accéder à l’une des 2^B entrées de la LUT implique un routage complexe. Pour palier à ce problème de taille, les échantillons précédents peuvent être sous-échantillonner/tronquer afin d’obtenir un adressage sur moins de bits, ou encore certains des bits peuvent être masquer selon une phase d’apprentissage liée au convertisseur à corriger [LASH].

Les résultats issus de ces techniques, bien qu’encourageant, sont mitigés. Dans [LSH05], l’auteur compare différentes configurations de la technique de masquage de bits [LSH04]. Cette technique semble être la plus prometteuse. Le tableau 2.1 reprend les valeurs de son analyse effectuée sur un convertisseur pipeline 10 bits fonctionnant à 20 MSPS. Ces valeurs représentent l’amélioration moyenne observée en dB pour différents paramètres sur l’ensemble de la bande de Nyquist.

K	adressage	SFDR	SINAD	THD
0	10bits	10.6	4.2	-10.0
1	20bits	12.4	2.6	-9.5
1	10bits	13.6	4.4	-10.6
1	5bits	11.3	4.1	-9.6
4	10bits	12.9	3.9	-9.3
4	18bits	20.3	5.4	-17.9

TAB. 2.1 – Amélioration des paramètres dynamiques en fonction des configurations [Lun03]

La première ligne correspond à une post-correction statique, les suivantes correspondent à diverses configurations liées à un masquage spécifique des bits d’adressage. L’auteur conclue que les performances obtenues sont étroitement liées à la configuration utilisée (valeur de K, et choix du masque d’adressage), et tente par la suite d’obtenir un outil d’optimisation de cette sélection. La conclusion de ses travaux sont illustrés sur la figure 2.6. La correction optimisée pour utiliser autant de mémoire que la correction statique est légèrement meilleure que cette dernière, mais le gain reste faible. Bien que prometteur, ce gain ne nous semble pas suffisant face à l’ajout de complexité du procédé de correction.

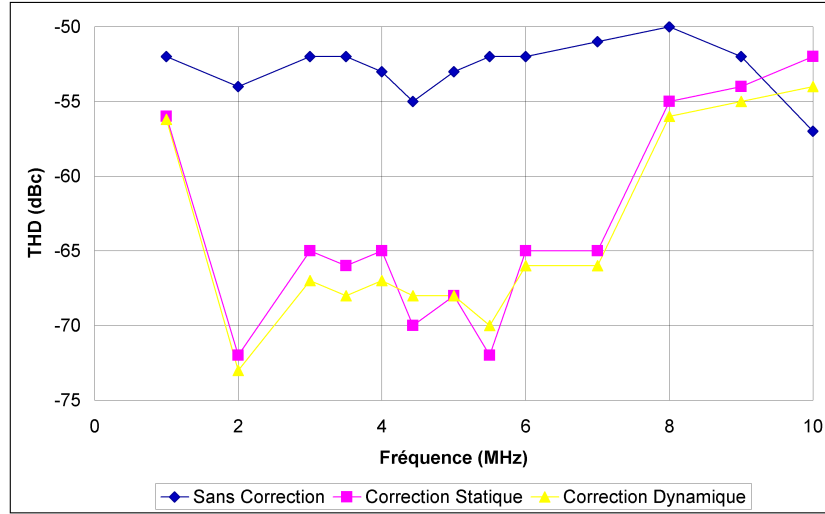


FIG. 2.6 – THD avec et sans correction (LUT calculée à $f = 3$ MHz) [Lun03]

Post-correction Plan de Phase

L'alternative à la correction d'Espace d'Etat, est celle dite de Plan de Phase (*Phase-Plane*). Le principe est similaire à la technique précédente si ce n'est que l'information supplémentaire traduit non plus l'état de l'échantillon précédent, mais la valeur de la pente du signal d'entrée [RI87, Mou89, Ber04]. La justification de cette technique provient d'un simple développement en séries de Taylor. En considérant le signal d'entrée $x(t)$ comme étant continu et différentiable, on peut alors décrire $x(t)$ au voisinage de l'instant $t = t_0$ par la formule suivante :

$$x(t) = x(t_0) + \sum_{k=1}^{\infty} a_k \cdot (t - t_0)^k \cdot \left. \frac{d^k x}{dt^k} \right|_{t=t_0} \quad (2.1)$$

Les travaux de Rebold et Irons [IHK91] montrent que le signal est décrit avec suffisamment de précision même lorsque seule la dérivée première est prise en compte. L'équation précédente devient alors une équation différentielle du premier ordre :

$$x(t) = x(t_0) + a_1 \cdot (t - t_0) \left. \frac{dx}{dt} \right|_{t=t_0} \quad (2.2)$$

Le système précédent dépend donc de deux variables dynamiques dont la dépendance temporelle peut être éliminée lorsque son évolution est étudiée dans l'espace des phases. Cette espace est défini par les deux variables dynamiques, qui dans notre cas, se révèlent être x et sa dérivée. Le signal x est donc décrit dynamiquement en fonction de sa fréquence, on justifie alors l'utilisation de l'estimation de la pente pour affiner la correction statique. Par conséquent, une mesure et/ou l'estimation du couple $(x[j], \frac{dx[j]}{dt})$ permet de retrouver l'erreur commise par le convertisseur. De la même façon que pour la post-correction d'Espace d'Etat, une table d'erreur à trois dimensions peut être décrite comme illustrée sur la figure 2.7.

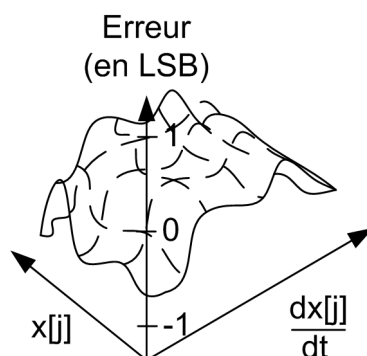


FIG. 2.7 – Plan de Phase à deux dimensions (K=1)

Une des contraintes est que la qualité de la correction est obtenue lorsque l'estimation de la dérivée se fait au voisinage immédiat du point d'échantillonnage : c'est à dire pour deux mesures effectuées à un intervalle de temps très petit. Dans le cadre des architectures classiques, l'estimation de la pente ne peut s'effectuer qu'à partir des échantillons issus de la conversion, or ces derniers sont tous distant d'une période d'échantillonnage bien trop grande pour obtenir une estimation de la pente suffisamment bonne. La précision de la correction est donc limitée, même si les résultats sont du même ordre de grandeur que la post-correction d'Espace d'Etat. La technique peut également être généralisée à l'ordre K en utilisant plus de dérivées [Tsi95].

2.2.3 Remplissage des Look-Up Table

Les trois techniques présentées précédemment réclament toutes l'utilisation d'une phase de calibration afin de remplir les valeurs des Look-Up Table. Cette phase est primordiale et nécessite souvent l'utilisation de signaux de références aux caractéristiques connues et maîtrisées.

Dans le cas d'une correction statique, un simple test par histogramme, obtenu à partir d'un stimulus sinusoïdal ou d'une rampe, suffit pour obtenir la courbe des non-linéarités intégrales [MotIIMS01]. Pour les deux autres techniques, il faut multiplier les signaux de référence [TLS⁺97] afin de couvrir un espace de correction le plus large possible.

2.2.4 Bilan de la compensation par table de correspondance

Les compensations par table de correspondance sont très simples à implémenter de façon interne au composant et fournissent des résultats qui montrent une amélioration significative des performances du convertisseur. Leur utilisation semble donc convenir à l'objectif fixé au chapitre précédent : compenser les non-linéarités sans augmenter de façon excessive la taille du composant. La différence de complexité entre les corrections statique et dynamique vis à vis du gain en performance, nous incite à nous limiter à l'utilisation d'une simple correction statique. De plus, la majorité des travaux de correction dynamique a été effectuée sur des architectures de type flash, or nous nous intéressons à des architectures de type à repliement ou pipeline, qui possèdent des non-linéarités plus complexes à étudier. Pour ce genre

d'architecture, les concepteurs utilisent déjà différentes techniques permettant de limiter les effets des non-linéarités. Ces techniques sont développées plus loin dans ce chapitre.

2.3 Méthodes basées sur la modélisation des Convertisseurs

La modélisation des convertisseurs permet d'obtenir une description analytique de leur comportement et donc d'estimer leur déviation par rapport au comportement idéal. Le convertisseur est considéré comme une "boîte noire" dont le signal d'entrée $x(t)$ est transformé par une opération $T[\cdot]$ en un signal de sortie $y(t)$ (figure 2.8(a)). Cette opération $T[\cdot]$ représente le système et dépend de l'architecture interne du convertisseur. Dans le cas idéal, cette opération se traduit par la fonction de transfert linéaire. L'ajout de non-linéarités statiques et dynamiques représentées par l'ajout de la fonction $\varepsilon[\cdot]$, impliquent une déformation de cette fonction de transfert comme indiqué précédemment (cf chap. 1). L'obtention d'une description de cette opération $T[\cdot] + \varepsilon[\cdot]$ non-parfaite permet d'inverser cette opération et par conséquent de la corriger. Ce principe est illustré sur la figure 2.8(b).

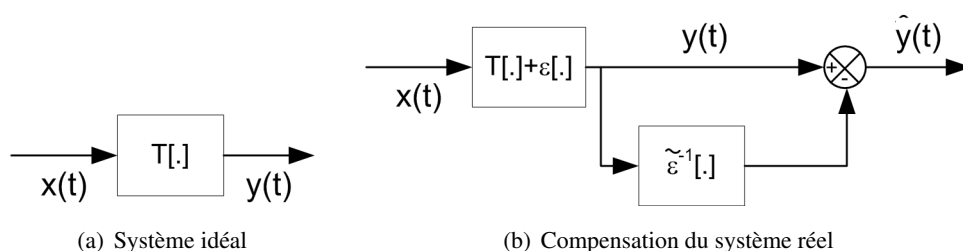


FIG. 2.8 – Modélisation et compensation du convertisseur

La modélisation consiste à décrire au mieux l'opération $T[\cdot] + \varepsilon[\cdot]$ en n'ayant que très peu d'information sur sa forme réelle. Ce type de méthode se nomme modélisation *a priori* du système. Dans le cas du convertisseur, les informations à notre disposition sont le signal d'entrée et les échantillons de sorties. Pour simplifier la modélisation, la non-linéarité et l'invariance temporelle de l'opération $T[\cdot]$ sont admises. Cette invariance traduit le fait que les effets temporels sur le composant, tel le vieillissement, sont négligés. Ainsi, lorsque le même signal d'entrée est présenté à des instants t différents à l'entrée du convertisseur, celui-ci fournit le même code de sortie.

Dans la section précédente, un développement en série de Taylor a été grossièrement utilisée pour décrire le comportement dynamique du convertisseur en fonction de la valeur du signal échantillonné et de la valeur de la pente à cet instant. Un développement similaire peut être utilisé afin de décrire le système représenté par l'opération $T[\cdot] + \varepsilon[\cdot]$, mais en tenant compte de la mémoire du système. Ce développement utilise principalement les séries de Volterra [Vol87], et leur utilisation dans le cadre de la modélisation qui nous intéresse ici est décrite dans les paragraphes suivants.

2.3.1 Les séries de Volterra

Dans la généralisation des approches d'Espace d'Etat et de Plan de Phase pour les corrections à l'aide des LUT, $K+1$ échantillons sont pris en compte pour obtenir la valeur corrective à appliquer. Cette prise en

compte traduit la mémoire du système de conversion. Le système est donc défini comme une opération non-linéaire, invariant dans le temps et à mémoire. Sous ces conditions, sa modélisation requiert l'utilisation des séries de Volterra. Ces dernières ont été développées en 1887 par Vito Volterra [Vol87] afin de généraliser la description en séries de Taylor des systèmes aux propriétés précitées. Sa formulation générale se présente sous la forme suivante :

$$y(t) = k_0 + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} k_n(\tau_1, \tau_2, \dots, \tau_n) x(t - \tau_1) x(t - \tau_2) \dots x(t - \tau_n) d\tau_1 d\tau_2 \dots d\tau_n \quad (2.3)$$

Cette description est similaire à celle des systèmes linéaires qui sont décrits par des coefficients h_i relatifs à la réponse impulsionnelle finie du système. Les variables $\{\tau_i\}$ représentent la mémoire du système, tandis que l'ensemble $\{k_i\}$ représente l'ensemble des noyaux d'ordre i de la série de Volterra. Ces noyaux peuvent être considérés comme la réponse impulsionnelle d'ordre élevé du système. La détermination de ces noyaux est très complexe à calculer car leur description fait intervenir un ensemble d'intégrales de convolution aux bornes infinies.

Pour les systèmes discrets, la forme générale précédente est adaptée selon la forme suivante :

$$y[n] = k_0 + \sum_{i=1}^{M_v} \sum_{m_1=0}^{N_v} \sum_{m_2=0}^{N_v} \dots \sum_{m_i=0}^{N_v} k_{i,m_1,m_2,\dots,m_i} \cdot \prod_{j=1}^i x[n - m_j] \quad (2.4)$$

M_v représente l'ordre maximal choisi du système, et N_v correspond à la valeur de la mémoire du système. Dans le cadre de la modélisation des convertisseurs, seuls des ordres faibles sont considérés $M_v = 3$ [BIHR] ou $M_v = 5$ [TLS⁺93, TL96], et la mémoire du système est également faible $N_v = K = 2$. Ce faible choix de paramètres se justifie par la complexité à obtenir les noyaux et à assurer la convergence de la série [TLS⁺96].

2.3.2 Identification du modèle

De nombreuses méthodes ont été utilisées pour identifier les paramètres du modèle de Volterra. Le travail mené par Wiener et exposé par Schetzen dans [Sch81], décline une nouvelle série de fonctions qui ont la propriété d'être orthogonales entre elles. Le passage des fonctions de Wiener aux fonctions de Volterra se fait alors assez simplement, fournissant une modélisation suffisamment précise du système [Tsi95]. Lee et Schetzen propose une autre alternative en considérant l'auto-corrélation [LS65] des signaux d'entrée et de sortie. Cette méthode est employée dans [MŠ] et fournit également des résultats corrects.

D'autres méthodes, proposées dans la littérature liée aux mathématiques, par exemple par Franz & Schölkopf, van Hemmen ou Korenberg, ont également été mises au point. La majorité d'entre elles nécessite l'utilisation d'un stimulus Blanc Gaussien [Sch74], même si des recherches ont été menées afin de se passer de cette contrainte forte [Sch81, KY79].

L'utilisation des séries de Volterra permet d'obtenir un modèle précis (au sens des moindres carrés) du système de conversion. A partir des échantillons d'entrée, la sortie peut être analysée et comparée à la sortie idéale. L'inversion du modèle permet alors de trouver la valeur corrective à appliquer pour obtenir un système compensé.

2.3.3 Inversion et compensation

Une fois l'obtention des noyaux de Volterra, l'inversion du système peut s'effectuer. Ce calcul n'est pas trivial et réclame une vérification de la convergence de la série inverse obtenue. Dans [TLS⁺97], un modèle d'ordre 5 est inversé et utilisé pour compenser les non-linéarités d'un modèle d'échantillonneur. Deux résultats principaux ressortent de leurs essais :

- a. le modèle de Volterra utilisé reproduit bien les non-linéarités du modèle de test.
- b. la compensation permet d'atténuer l'effet de ces non-linéarités.

Cependant, un ordre et une mémoire de taille plus élevés semblent être nécessaires pour obtenir une compensation plus performante que celle obtenue par les méthodes avec Look-Up Table. Dans [TL96], les deux méthodes sont comparées et la conclusion appuie ce dernier point.

L'intérêt de la compensation à inversion de modèle s'avère donc très limité du fait de la complexité des calculs nécessaires à l'obtention du modèle et de son inverse. De plus la nature des stimuli de tests est également une contrainte supplémentaire. Des travaux plus récents [MS02] permettent de réduire la complexité et d'utiliser d'autres signaux, mais les techniques ne sont pas adaptables à tout les types de convertisseurs.

En étudiant de près les architectures utilisées, de nouvelles méthodes de compensation ont donc été élaborées. C'est l'objet de la section suivante.

2.4 Méthodes de compensation interne

Les méthodes de compensation liées à l'architecture regroupent deux grands ensembles : la calibration agissant sur les organes analogiques des convertisseurs, et la compensation numérique qui utilise les informations issues des conversions internes pour corriger les dérives des composants analogiques. Chaque ensemble requiert une connaissance poussée de l'architecture interne du circuit.

2.4.1 La technique du dither

Dans les années 40, l'aviation britannique se rendit compte que les systèmes de navigation, entièrement mécanique à l'époque, étaient difficiles à calibrer au sol, alors que dans l'air, les vibrations du moteur semblaient rendre plus fluide les divers mouvements des engrenages et autres leviers. Pour retrouver les conditions du vol lors des phases de calibration, les ingénieurs rajoutèrent des moteurs vibrant autour des pièces à calibrer. Cette opération avait pour effet de rendre prévisible le comportement de la mécanique, alors qu'au repos, des engrenages pouvaient aléatoirement se bloquer. Ils nommèrent cette opération dither.

En traitement du signal, l'opération de dithering [Sch64] se traduit par l'ajout d'un signal aléatoire connu, afin de fluidifier l'effet de la quantification et de minimiser l'impact des erreurs de linéarité. La figure 2.9 illustre le fonctionnement de la technique. La figure de gauche représente le signal quantifié sur 5 bits et son spectre associé. Le signal de droite représente le même signal quantifié après l'ajout d'un bruit blanc de l'ordre du LSB sur le signal analogique. Le spectre du premier signal quantifié présente de

nombreuses raies harmoniques tandis que le spectre du signal quantifié après ajout du signal de dither, en a atténué une grande quantité (figure 2.10). Cependant, le plancher de bruit est plus élevé que celui de la simple quantification. La technique de dither est souvent utilisée dans les applications audio, où un bruit plus élevé est préférable à la présence d’harmoniques audibles.

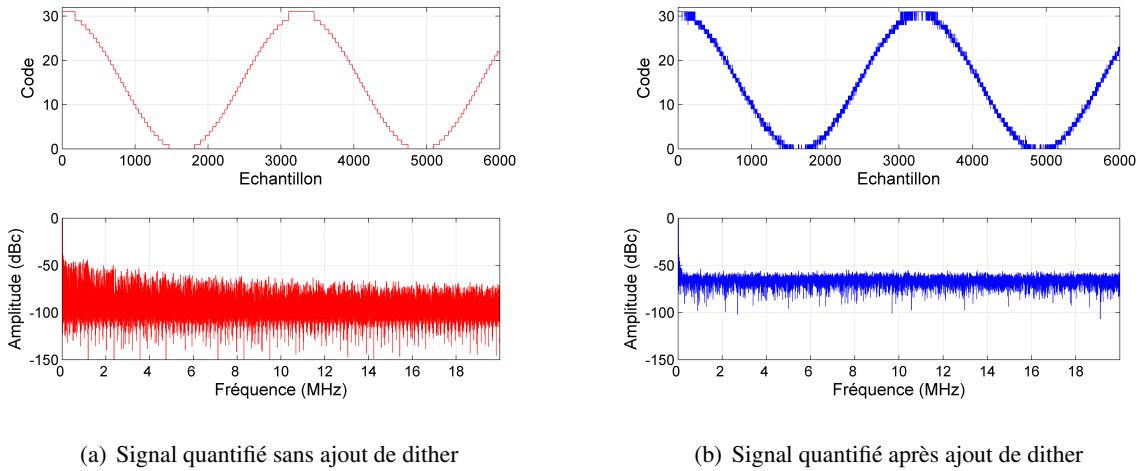


FIG. 2.9 – Allure temporelle et spectre des signaux sans et avec dither

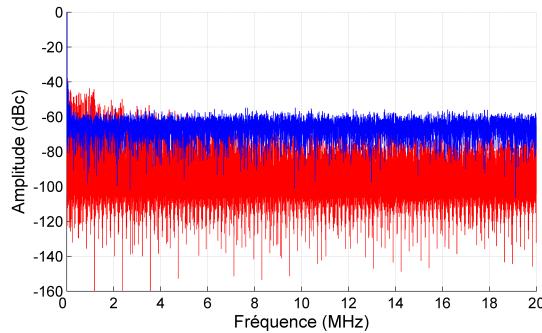


FIG. 2.10 – Superposition des spectres sans et avec dither

2.4.2 Calibration des capacités pour un convertisseur Pipeline

Les convertisseurs de type pipeline ont été partiellement décrit dans le chapitre précédent. La représentation classique d’un étage de conversion est décrite sur la figure 2.11(a). Le sous-CAN fournit un code sur m bits, qui est envoyé dans le CNA multiplicateur (MDAC), afin de créer le signal résiduel. Un descriptif plus détaillé est fourni dans la figure 2.11(b).

Il s’agit d’un amplificateur opérationnel et d’un assemblage de capacités reliées à différentes références. La capacité C_F est la capacité de retour du signal, tandis que les capacités $C_{S,i}$ sont dites d’échantillonnage. Le fonctionnement ne sera pas décrit plus en détails, ces figures sont fournies pour mettre en évidence les problèmes de non linéarité de l’étage. Le gain de l’amplificateur, et la mauvaise adaptation

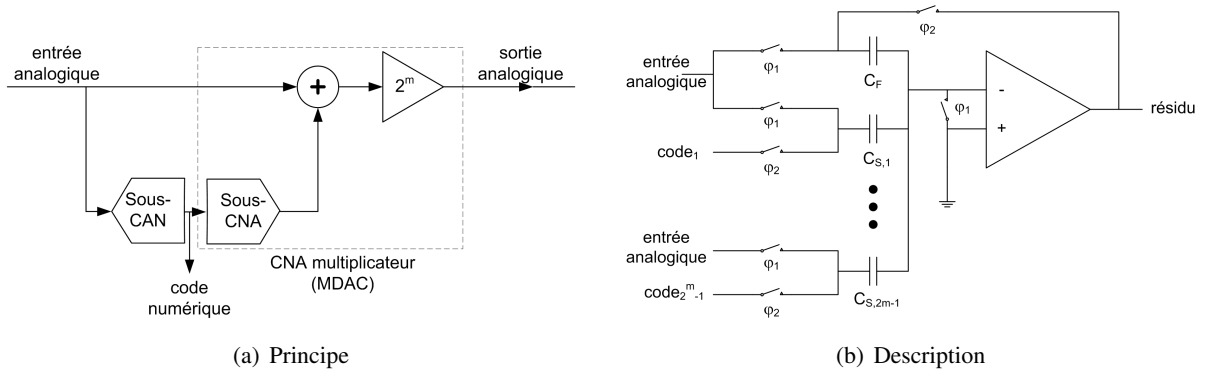


FIG. 2.11 – Principe et description d’un CNA multiplicateur

des capacités entre elles dégradent la linéarité de l’ensemble. Pour remédier aux problèmes d’adaptation, différentes techniques sont proposées [TSH06, SHT⁺00]. Globalement, l’idée est de procéder à une permutation pseudo-aléatoire des capacités de retour et d’échantillonnage afin de moyennner les effets récurrents de ces mauvaises adaptations. Les permutations sont effectuées de façon pseudo-aléatoire à partir de signaux de commandes numérique représentés par les variables φ_1 et φ_2 [TSH06]. A l’instar de la technique de dither, cette calibration permet d’atténuer la valeurs des raies harmoniques tout en augmentant légèrement le plancher de bruit, et donc elle améliore le *SFDR* du composant.

Grâce aux modes de calibration, les mauvaises adaptations peuvent être calculées de façon numérique à partir de blocs digitaux simples [HZHvR06]. De la même manière, le gain de l’amplificateur est ajusté pour s’adapter aux étages en aval.

Les effets de ces corrections sont illustrés sur la figure 2.12. Les figures de gauche représentent les courbes de non-linéarités au fur et à mesure des corrections de gain des différents étages et des compensations des erreurs d’appariement des capacités. Les effets de ces non-linéarités apparaissent sur les spectres placés en vis-à-vis des courbes de NLI. Les raies harmoniques s’atténuent en même temps que les non-linéarités.

Sans processus de corrections des déviations des composants, la conversion est beaucoup trop éloignée du cas idéal. Avec ces processus, la conversion est grandement améliorée. L’utilisation de calibrations est donc obligatoire dans la conception d’architectures rapides et fiables.

2.4.3 Correction numérique adaptative dans un convertisseur $\Sigma\Delta$

Un autre exemple de correction numérique s’observe dans la conception des architectures de type $\Sigma\Delta$ cascades (MASH) [CT92, BW88]. Le convertisseur $\Sigma\Delta$ permet d’obtenir facilement une résolution élevée du fait de son opération de mise en forme du bruit de quantification, mais l’utilisation du suréchantillonnage implique une bande de fréquence du signal d’entrée assez étroite. Le modulateur $\Sigma\Delta$ est décrit de façon analogique par la figure 2.13(a), mais pour étudier la compensation des imperfections des capacités et de l’amplificateur, un modèle linéaire discret est utilisé (figure 2.13(b)), avec :

$$a = \frac{\frac{C_1}{C_2}}{1 + \frac{1}{A_{DC}} \left(1 + \frac{C_1}{C_2} \right)} \quad (2.5)$$

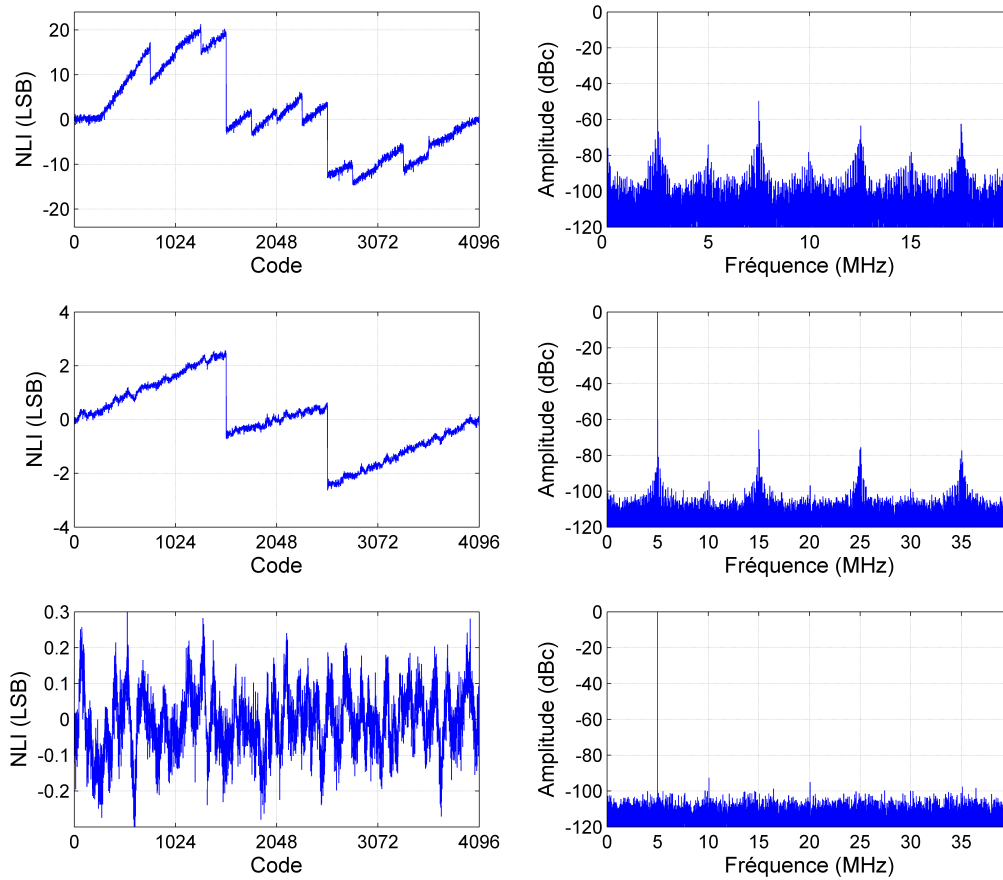


FIG. 2.12 – Effet de la calibration sur la courbe des non linéarités

et

$$p = \frac{1 + \frac{1}{A_{DC}}}{1 + \frac{1}{A_{DC}} \left(1 + \frac{C_1}{C_2}\right)} \quad (2.6)$$

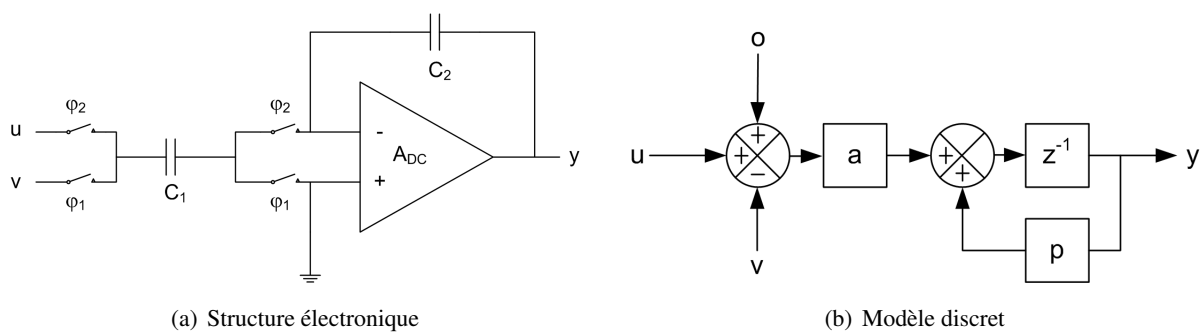


FIG. 2.13 – Modélisation d'un modulateur $\Sigma\Delta$

2.5. CONCLUSION

Cette modélisation illustre l'influence de la mauvaise adaptation des capacités C_1 et C_2 , du gain de l'amplificateur A_{DC} , et de la présence d'un offset o issu des décalages propres à l'amplificateur, et des différents bruits (thermique et $1/f$).

A partir de cette nouvelle écriture, une fonction de transfert du modulateur peut être définie, et de manière similaire à la correction par inversion de modèle, cette fonction de transfert peut être rendue idéale. Une des façons de rendre idéale cette fonction, est d'utiliser des filtres à réponse impulsionnelle finie (FIR) [HIUK86, LS90]. Les coefficients des filtres (ordre faible) sont calculés soit à partir d'un signal de test injecté dans le processus de conversion [KSW⁺00], soit lors d'une phase de calibration sans stimuli [CT00]. Dans les deux cas, un filtrage adaptatif basé sur le critère des moindres carrés permet d'optimiser la valeur de ces coefficients [CT96, AKTS92].

Les conclusions de ces procédés annoncent un gain d'une vingtaine de dB sur le *SINAD*, permettant ainsi de gagner plusieurs bits de résolution.

2.5 Conclusion

Au travers de cet état de l'art non exhaustif, plusieurs méthodes aux propriétés diverses et variées ont été étudiées. Le tableau 2.2 synthétise les avantages et inconvénients des principales méthodes présentées.

Méthode	Complexité	Temps de Calcul	Inconvénient
LUT 1D	faible	faible	dépend de la fréquence
LUT nD	moyenne	moyen	taille mémoire importante
Volterra	élevée	élevé	complexité des calculs
Dither	faible	faible	élévation du plancher de bruit
Calibration	faible	faible	liée au design

TAB. 2.2 – Comparaison des méthodes de compensation

La plupart des méthodes ne sont pas concurrentes et peuvent être appliquées au sein d'une même correction. En effet, la calibration des capacités peut tout d'abord être effectuée, et une correction par LUT statique peut ensuite être appliquée. Les diverses méthodes exposées ci-dessus nous permettent de sélectionner une voie pour la post-correction que l'on souhaite utiliser. Les méthodes à LookUp Table semblent fonctionner de manière suffisamment précise vis à vis de l'amélioration de performances visées. Par contre, la complexité supplémentaire apportée par la multiplication des dimensions des tables ne nous semble pas justifiée pour cette première étude. Nous préférons donc nous limiter à une correction à une dimension. Le mode de remplissage habituelle des tables de correspondance se fait par la méthode classique de l'histogramme, cette méthode et ses limitations sont exposées dans le chapitre suivant. Nous souhaitons utiliser une modélisation de convertisseurs pour coller au plus près à leurs non-linéarités dynamiques, le chapitre suivant expose donc une nouvelle méthode de remplissage de la table de correspondance.

Chapitre 3

Extraction des non-linéarités

La technique de compensation par table de correspondance unidimensionnelle a été sélectionnée afin de répondre à nos objectifs. Selon l'option de compensation offline ou online choisie, les ressources disponibles pour l'extraction des non-linéarités sont différentes. Dans le premier cas, les ressources sont illimitées, et dans l'autre, elles sont restreintes au maximum afin d'être intégrables au sein du composant. Ce chapitre expose deux principes d'extraction des non-linéarités adaptés à ces deux besoins.

3.1 Introduction

Pour remplir la table de correspondance, il faut pouvoir extraire les non linéarités du convertisseur. Différentes techniques s'offrent à l'utilisateur. Certaines sont basées sur une analyse statistique des données de sortie, d'autres sont basées sur une méthode analytique. Ces deux approches sont expliquées dans les paragraphes suivants et le chapitre conclut sur l'expression des besoins pour la compensation et du choix de la méthode de remplissage de la LUT.

3.2 Extraction basée sur une approche statistique

Pour évaluer la fonction de transfert, et donc les non-linéarités, il est nécessaire de détecter toutes les tensions de transition V_i du convertisseur. L'idéal est de pouvoir faire évoluer un signal d'entrée de type rampe et de noter la valeur du signal électrique lors des transitions de bits dans le code de sortie. En pratique, l'obtention d'un signal rampe précis est compliqué à mettre en oeuvre du fait de la présence de nombreux bruits tout au long de la chaîne de conversion : le signal n'est alors plus linéaire. Cette présence de bruit rend donc hasardeuse la mesure de la transition. Pour palier à ces phénomènes, le moyennage de l'ensemble des valeurs est utilisé. Ainsi, le signal d'entrée n'est plus constitué d'un seul motif (une rampe, ou un lobe de sinusoïde), mais d'une répétition de ces motifs. En augmentant ce nombre de motifs, l'effet du bruit devient maîtrisable, dans le sens où l'on peut déterminer mathématiquement un niveau de confiance u lié à une tolérance B sur la valeur mesurée. L'utilisation de ce processus de moyennage justifie l'appellation d'approche statistique. Les paragraphes suivants exposent l'utilisation de la technique dite

3.2. EXTRACTION BASÉE SUR UNE APPROCHE STATISTIQUE

classique, celle par histogramme. Les avantages et les inconvénients de cette technique sont exposés au fur et à mesure de son explication.

3.2.1 Méthode classique de test par histogramme

La méthode classique [MotIIMS01] de mesure des non-linéarités d'un convertisseur est celle dite de l'Histogramme [BSR86, Mah87, Pac, MCD⁺00]. Elle consiste en l'application à l'entrée du convertisseur, d'un signal aux propriétés statistiques connues, et dont l'influence de la quantification du convertisseur idéal produit une représentation statistique prévisible de la fréquence d'apparition des codes. Par exemple, l'envoi d'un signal analogique de type rampe fournit, dans le cas idéal, une densité d'apparition de code constante liée à la pente de la rampe et à la fréquence d'échantillonnage. La figure 3.1 illustre ce propos : la valeur de la pente par rapport à la fréquence d'échantillonnage permet d'obtenir deux points par code, ce qui se traduit par une densité d'apparition de code constante de valeur 2.

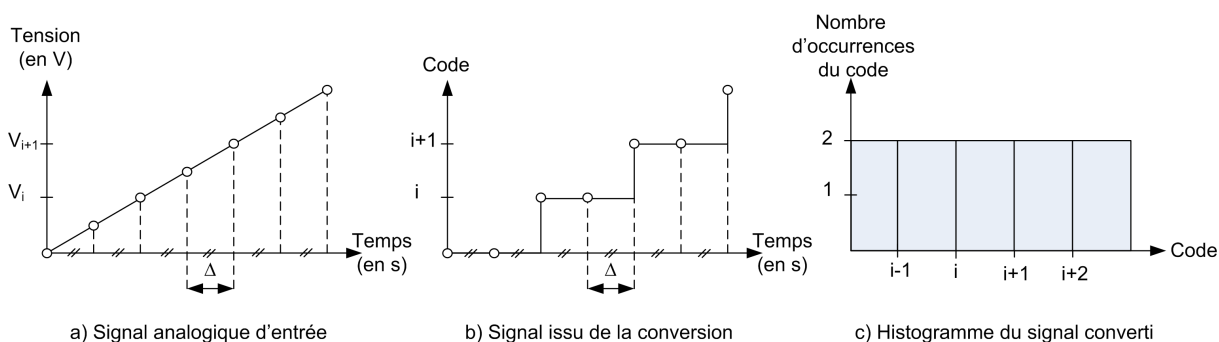


FIG. 3.1 – Densité de code dans le cas idéal

Par contre, l'ajout de l'influence du bruit sur le signal d'entrée et l'introduction du jitter (traduit par δ_1 et δ_2) perturbe la densité d'apparition de code indépendamment des non-linéarités du convertisseur. La figure 3.2 illustre ce dernier cas. Il faut donc prendre un très grand nombre de point afin de minimiser l'impact des différents bruits.

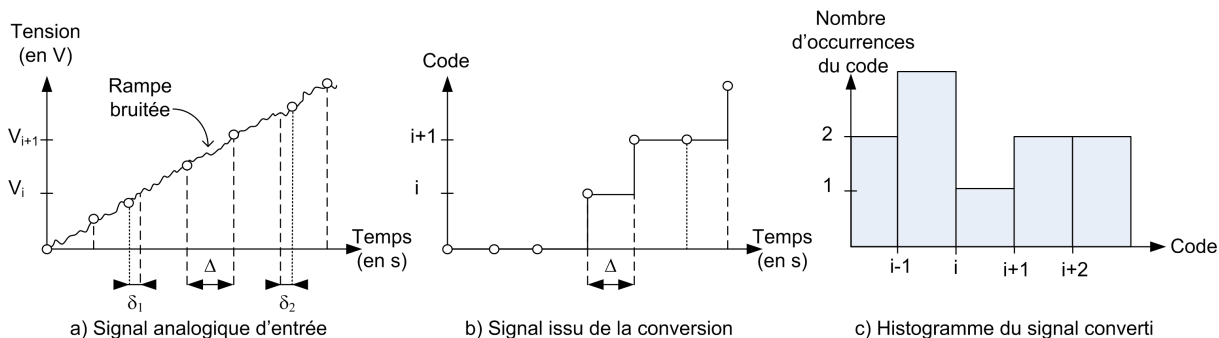


FIG. 3.2 – Densité de code dans le cas réel

Le terme histogramme est préféré, pour sa simplicité, à celui de densité d'apparition de code. Comme indiqué précédemment, l'utilisation d'un seul motif n'est pas suffisant pour s'assurer que les non-linéarités présentes dans l'histogramme soient effectivement celles du convertisseur, et non, dues aux imperfections de la mesure. Dans [Bla94], une analyse détaillée est présentée afin d'obtenir une formulation mathématique du nombre d'échantillons nécessaires pour une représentation fiable de la mesure. Par défaut un signal sinusoïdal est utilisé, celui-ci étant plus facile à produire en laboratoire. Selon cette étude, le nombre de points nécessaires S pour obtenir une mesure avec un niveau de confiance $1 - u$ lié à une tolérance B est donnée par la formule suivante :

$$S \approx \left(\frac{\pi}{2}\right) 2^n \left(\frac{\sqrt{2} \operatorname{erfc}^{-1} \left(1 - (1 - u)^{2^{-n}} \right)}{B} \right) \quad (3.1)$$

où $\operatorname{erfc}^{-1}(\cdot)$ est l'inverse de la fonction d'erreur complémentaire, et n est la résolution du convertisseur. Pour une tolérance de 25% (en terme de largeur de code) et un niveau de confiance de 99%, si la variance du bruit d'entrée est de l'ordre de 1 LSB (rms), et que les acquisitions sont effectuées par un assemblage d'acquisitions cohérentes de taille 2^n ([Mah87]) mais de phase à l'origine aléatoire, alors 8Méga-points sont nécessaires pour l'estimation des non linéarités d'un convertisseur de résolution $n = 14$ bits. En respectant ces contraintes, l'obtention de l'histogramme nous permet, avec la tolérance et la confiance choisies, de calculer les non-linéarités de façon fiable. Son principe de fonctionnement est expliqué en annexe A de ce document.

3.2.2 Utilisation en vue de la compensation

Lors de la phase de test d'un convertisseur, une mesure de la NLI est effectuée systématiquement à partir de la méthode par histogramme. L'utilisation du résultat de cette mesure pourrait donc être utilisée pour remplir la table de correspondance sans ajouter de coût supplémentaire au prix de revient du composant. Malheureusement, les conditions de la mesure ne sont pas toujours adaptées au calcul optimal de la table de correspondance (voir chapitre suivant), et une seconde mesure doit alors être effectuée. Cette nouvelle mesure ajoute un surcoût significatif lié au nombre élevé d'échantillons nécessaires à l'obtention d'une mesure fiable. Les courbes de la figure 3.3 représentent l'allure de l'estimation de la NLI en fonction du nombre de points utilisés. Les valeurs obtenues pour les faibles nombres de points sont beaucoup trop éloignées des valeurs réelles et ne peuvent donc être exploitables en tant que valeurs de remplissage pour la table de correspondance.

De plus, dans le cas de la compensation *online*, l'utilisation d'un grand nombre d'échantillons est également un facteur pénalisant. En effet, pour les convertisseurs à haute vitesse, le processus de calcul de l'histogramme ne peut s'effectuer en temps réel et impose un stockage des échantillons avant leur traitement. Ce stockage (8Méga-points pour un convertisseur de résolution 12bits) engendre un surcoût important en terme de surface de silicium. Il faudrait donc pouvoir réduire le nombre de points tout en conservant une précision suffisante. La méthode par histogramme ne répond donc pas à ce besoin.

L'approche analytique tente de résoudre ce problème, afin de réduire le nombre de points nécessaires tout en conservant une précision suffisante sur le calcul des non-linéarités.

3.3. EXTRACTION À PARTIR DU SPECTRE DE SORTIE DU CAN

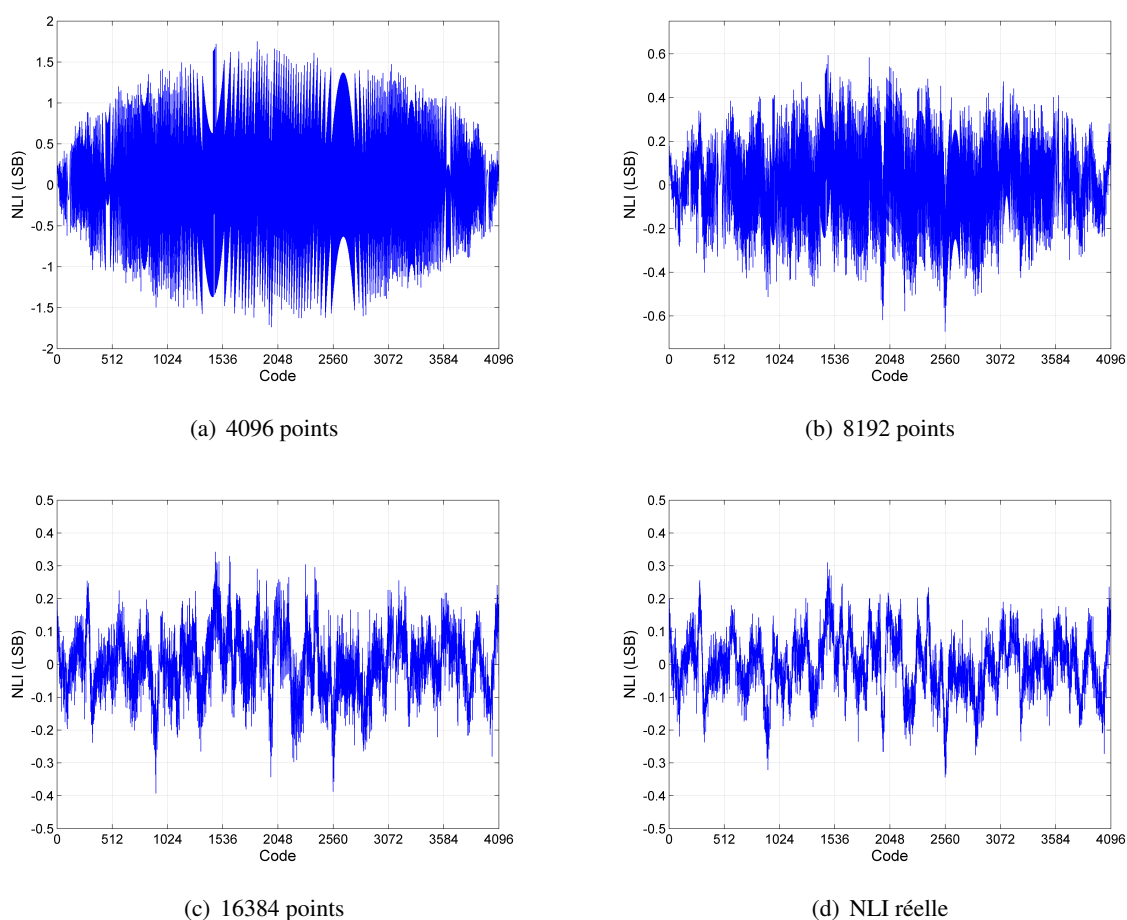


FIG. 3.3 – Estimation de la courbe des NLI en fonction du nombre de points utilisés

3.3 Extraction à partir du spectre de sortie du CAN

L'approche statistique mentionnée dans la section précédente fait partie des techniques dites statiques de caractérisation d'un convertisseur. Comme leurs noms l'indiquent, ces méthodes permettent de faire ressortir les erreurs statiques (cf chap1) du composant sous test. L'extraction des paramètres dynamiques (cf chap1) est réalisée par une exploitation fréquentielle des données converties. En appliquant une Transformée de Fourier Rapide (ou Fast Fourier Transform) [CT65] sur les données acquises, l'utilisateur est à même d'extraire les paramètres tels que le *SNR*, le *SFDR* ou la *THD* [MotIIMS01]. L'utilisation de la FFT réclame une acquisition cohérente [Mah87] d'échantillons dont le nombre nécessaire pour l'obtention d'une mesure fiable est de l'ordre de 4×2^n afin de faciliter les calculs numériques [CT65]. Ce faible nombre d'échantillons (65kilo-points pour un convertisseur de résolution $n = 14$ bits) permet d'obtenir un temps de test très inférieur à celui du test par histogramme (8Méga-points nécessaires pour la même résolution de CAN). Le temps de test industriel étant une contrainte intégrée au prix de revient du composant testé, il convient de le réduire au maximum afin d'améliorer la rentabilité du produit. L'idéal serait donc de rassembler les deux mesures en une seule tout en gardant le même taux de couverture de test.

De nombreux travaux [BCA⁺04, Xu99, KBCJ06] font état de procédés tentant de rassembler les ex-

tractions des paramètres statiques à partir du test dynamique par FFT. Par exemple, dans [BCA⁺30], les auteurs proposent une méthode de test par apprentissage, qui permet de trier les convertisseurs ne correspondant pas aux spécifications statiques à partir de signatures dynamiques mesurées sur le spectre du signal converti. Cette technique se rapproche d'une pratique statistique dans le sens où la phase d'apprentissage s'effectue à partir de mesures statiques effectuées sur une population connue, dans le but d'extrapoler le comportement de l'ensemble des convertisseurs sous test. Bien que cette méthode semble efficace pour tester les CAN, elle n'est absolument pas utilisable pour évaluer les non-linéarités en vue d'une compensation.

D'autres travaux tentent quant à eux, d'intégrer les non-linéarités sous forme d'un modèle mathématique dans la chaîne de conversion, afin d'observer l'effet de ce modèle sur le spectre de sortie [AAG⁺02]. L'idée étant d'inverser ensuite ce modèle pour remonter aux non-linéarités à partir de mesures effectuées sur le spectre. Ce principe, s'il fonctionne, permettrait dans le cas de la compensation, d'avoir une extraction des non-linéarités à partir d'un faible nombre de points et donc rendrait viable l'option d'une compensation *online*. Les paragraphes suivants exposent deux méthodes de modélisation différentes des non-linéarités : une modélisation polynomiale, et une nouvelle modélisation fréquentielle développée en interne [JF07] et optimisée pour nos besoins. Leurs avantages et inconvénients sont discutés par la suite.

3.3.1 Modélisation des non-linéarités par un polynôme

Dans [BBGdE98, FNCS04, CJ], les auteurs décrivent le convertisseur comme étant une "boîte noire", où l'opération de conversion se traduit par la fonction $g(\cdot)$. La figure 3.4 illustre ce principe.

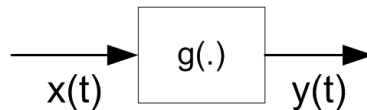


FIG. 3.4 – Conversion

Cette opération $g(\cdot)$ est une fonction statique non linéaire qui transforme un signal sinusoïdal du type $x(t) = V \cdot \cos(\omega t) + C$, en un signal de sortie de la forme :

$$y(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) \quad (3.2)$$

Cette dernière représentation est issue d'une simple décomposition en série de Fourier. Une des contraintes fortes utilisées dans cette modélisation provient du fait que le convertisseur est considéré comme étant invariant temporellement et sans mémoire. Cette contrainte permet d'affirmer que toutes les harmoniques sont parfaitement en phase, ce qui n'introduit pas de termes en sinus dans la décomposition en séries de Fourier. Cette hypothèse peut être considérée comme vraie dans le cas de signaux d'entrée à faible fréquence, et pour une fréquence d'échantillonnage également peu élevée. L'utilisation d'un étage d'entrée tampon de type *buffer* ou un E/B permettent également d'assurer la validité de ces hypothèses. Comme ces contraintes correspondent aux besoins d'une mesure statique des non-linéarités, leurs mis en oeuvre n'est pas un réel obstacle à l'utilisation de la modélisation. Pour vérifier si ces hypothèses sont toujours

3.3. EXTRACTION À PARTIR DU SPECTRE DE SORTIE DU CAN

validées, l'utilisateur doit vérifier dans le spectre complexe issu de la transformation de Fourier que toutes les harmoniques ont une phase proportionnelle.

La modélisation polynomiale de la fonction $g(\cdot)$ est effectuée à l'aide des polynômes de Chebyshev du premier ordre. Ces polynômes $C_n(\cdot)$ d'ordre n sont orthogonaux entre eux et vérifient la relation d'identité suivante :

$$C_n(\cos\theta) = \cos(n\theta) \quad (3.3)$$

En les pondérant par la fonction $\sqrt{1-x^2}$, ils forment la base d'un espace vectoriel orthonormal, et toute fonction $f(\cdot)$ peut alors être décrite dans cette base sous la forme :

$$f(x) = \frac{c_0}{2} + \sum_{n=1}^{\infty} c_n C_n(x) \quad (3.4)$$

où les coefficients c_n sont décrits par :

$$c_n = \frac{2}{\pi} \int_{-1}^1 \frac{f(x)C_n(x)}{\sqrt{1-x^2}} dx \quad (3.5)$$

Par identification entre 3.2 et 3.3 et en posant $z = \cos(\omega t)$, on obtient la nouvelle expression de a_n :

$$a_n = \frac{2}{\pi} \int_{-1}^1 \frac{g(Vz+C)C_n(z)}{\sqrt{1-z^2}} dz \quad (3.6)$$

En substituant cette expression dans 3.2, il apparaît clairement que la fonction $g(\cdot)$ peut être décrite comme une somme de polynômes de Chebyshev :

$$y(t) = g(x(t)) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n C_n\left(\frac{x(t)-C}{V}\right) \quad (3.7)$$

Implémentation du modèle polynomial

A partir de cette modélisation, le polynôme décrivant $g(\cdot)$ peut être totalement défini et donc comparé à la forme idéale qui ne possède pas de non-linéarités. L'obtention des coefficients de cette modélisation s'effectue à partir de l'équation 3.2 par identification des constantes a_n . Pour cela, il suffit de réaliser une transformée de Fourier rapide sur le signal de sortie, en respectant les contraintes de cohérence qui permettent d'obtenir une mesure représentative des composantes fréquentielles du signal. Si la fréquence d'échantillonnage est nommée f_{ech} , et la fréquence d'entrée f_{in} , alors il existe une relation dite de cohérence entre le nombre de points d'échantillons N_s et le nombre de cycles M du signal d'entrée présents dans l'enregistrement des données. Cette relation s'exprime sous la forme :

$$M \times f_{ech} = N_s \times f_{in} \quad (3.8)$$

Il faut de plus, que M et N_s soient premiers entre eux, et pour pouvoir utiliser la FFT [CT65], N_s est choisi comme étant une puissance de 2. Cette relation permet d'assurer l'unicité des points d'échantillonnage. L'opération de Transformée de Fourier rassemble l'information fréquentielle du signal dans des intervalles de fréquences d'égale largeur $\frac{f_{ech}}{N_s}$ Hertz. Toute l'énergie portée par un signal de fréquence appartenant à un intervalle donné, se retrouve additionnée à l'énergie de cet intervalle. Ainsi l'ensemble

des intervalles, appelé spectre complexe, représente la répartition fréquentielle de l'énergie du signal. Pour un signal sinusoïdal pur, toute l'énergie du signal est portée par la fréquence du signal. Dans le cas d'un signal sinusoïdal distordu, l'énergie est répartie sur la fréquence fondamentale et sur les fréquences harmoniques. Dans le cas d'un convertisseur, cette distorsion est en grande partie liée aux non-linéarités du composant. Il semble donc logique qu'à partir de la représentation fréquentielle, les non-linéarités puissent être extraites.

Le calcul de l'énergie des intervalles $Y(i)$ se fait à partir de la formule suivante :

$$Y(i) = \sum_{k=0}^{N_s-1} y\left(\frac{k}{f_{ech}}\right) e^{-\frac{j2\pi ik}{N_s}} \quad i = 0, \dots, N_s - 1 \quad (3.9)$$

Ces valeurs sont ensuite mises en relation avec la décomposition en série de Fourier de l'équation 3.2 à l'aide des formules suivantes :

$$a_0 = \frac{Y(0)}{N_s} \quad a_n = \frac{2Y(n \times M \bmod N_s)}{N_s} \quad n = 1, \dots, H_{max} \quad (3.10)$$

où H_{max} correspond à un nombre arbitrairement choisi correspondant à l'ordre de la dernière harmonique dont l'énergie est influencée par la distorsion. Ce calcul permet alors, à partir de 3.7, de retrouver l'équation du polynôme représentant $g(\cdot)$. La figure 3.5 représente la comparaison des non-linéarités d'un convertisseur 8bits, mesurées à partir de 256k échantillons, par la méthode par histogramme (trait fin) et calculées à partir d'un spectre complexe obtenue à partir de 8192 échantillons. Le polynôme parvient à reproduire la forme globale des non-linéarités, mais du fait de la limitation de l'ordre H_{max} du modèle, il ne parvient pas à reproduire les variations brusques qui caractérisent les courbes de NLI.

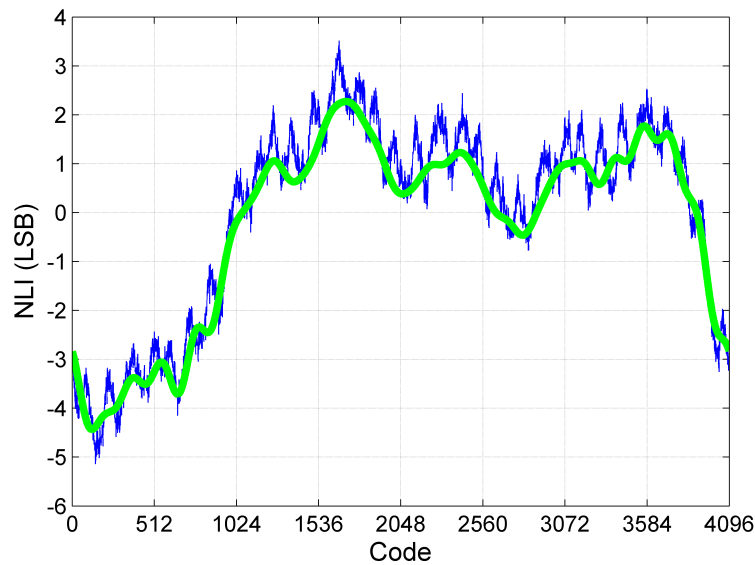


FIG. 3.5 – INL d'un 12bits et polynôme associé avec $H_{max} = 50$

On constate donc que les polynômes de Chebyshev réussissent à fournir la forme globale des non-linéarités, mais de part la nature erratique des courbes de non-linéarités, l'ordre limité du polynôme final est un obstacle à la reproduction de ces variations brusques. Augmenter l'ordre H_{max} n'est pas une

solution, car les calculs numériques deviennent alors imprécis et instables car les valeurs mesurées sont noyées dans le bruit de quantification. Cette approche doit donc être considérée comme une extraction avec moyennage de la courbe de non-linéarités. Cette forme globale peut éventuellement suffire pour la compensation, mais nous proposons une autre modélisation [JF07] afin de reproduire avec plus de précision ces variations. Cette technique est exposée dans les paragraphes suivants.

3.3.2 Modélisation des non-linéarités par série de Fourier

La description polynomiale n'est donc pas adaptée pour traduire les brusques variations présentes dans la courbe représentative des non-linéarités [KBJC05]. Une autre description doit donc être utilisée : en rendant périodique la fonction traduisant ces non-linéarités, la description naturelle qui nous semble s'imposer est celle des séries de Fourier.

En traduisant la conversion idéale d'un signal sinusoïdal pur du type $x(t) = V \cos(2\pi f_{int} t + \theta_0) + C$, et en respectant la relation de cohérence énoncée par l'équation 3.8, on obtient un signal quantifié idéal $u[l]$ qui peut être exprimé sous la forme suivante (en unité LSB) :

$$u[l] = 2^n \left(\frac{V}{PE} \right) \cos(\theta_l) + 2^n \left(\frac{C}{PE} \right) + q(l) \quad (3.11)$$

où le bruit de quantification est représenté par $q(l)$ et la phase instantanée θ_l est fournie par l'expression :

$$\theta_l = 2\pi \left(\frac{M}{N_s} \right) l + \theta_0 \quad (3.12)$$

La conversion non idéale, c'est à dire celle qui tient compte des non-linéarités du composant, peut s'exprimer par l'addition d'une fonction de distorsion $\Phi(\cdot)$, comme indiqué sur la figure 3.6.

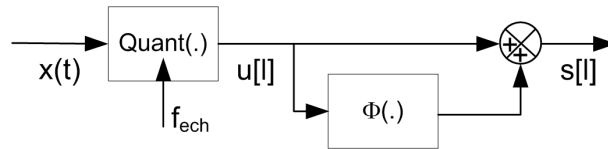


FIG. 3.6 – Conversion non idéale

Le signal de sortie $s[l]$ contenant le signal quantifié $u[l]$ et les distorsions introduites par $\Phi(\cdot)$ s'exprime donc par :

$$s[l] = u[l] + \Phi(u[l]) \quad (3.13)$$

C'est la fonction de distorsion $\Phi(\cdot)$ qu'il nous faut modéliser à l'aide des séries de Fourier. Pour cela, si $NLI(\cdot)$ représente la fonction du convertisseur qui à chaque code $x \in [0 \dots 2^n - 1]$ lui associe la non-linéarité $INL(x)$ en LSB, alors $\Phi(\cdot)$ est décrite par les équations suivantes :

$$\Phi(x) = NLI(x), \quad 0 \leq x \leq 2^n - 1 \quad (3.14)$$

$$\Phi(x + p \cdot 2^n) = NLI(x), \quad \text{pour tout } p \in Z \quad (3.15)$$

Cette fonction répond alors aux besoins de continuité et de périodicité pour être décomposable en séries de Fourier. Le développement en séries de Fourier de la fonction $\Phi(\cdot)$ s'écrit alors sous la forme suivante :

$$\Phi(x) \approx \frac{a_0}{2} + \sum_{k=1}^{2^n-1} \left[a_k \cos\left(2\pi \frac{k}{2^n} x\right) + b_k \sin\left(2\pi \frac{k}{2^n} x\right) \right] \quad (3.16)$$

avec les expressions usuelles des paramètres a_k et b_k :

$$a_k = \frac{1}{2^n} \sum_{x=0}^{2^n-1} NLI(x) \cos\left(2\pi \frac{k}{2^n} x\right) \quad k \in \{0, 1, \dots, 2^n - 1\} \quad (3.17)$$

$$b_k = \frac{1}{2^n} \sum_{x=0}^{2^n-1} NLI(x) \sin\left(2\pi \frac{k}{2^n} x\right) \quad k \in \{1, \dots, 2^n - 1\} \quad (3.18)$$

Cette décomposition peut désormais être introduite dans l'expression du signal distordu $s[\cdot]$:

$$s[l] = u[l] + \frac{a_0}{2} + \sum_{k=1}^{2^n-1} \left[a_k \cos\left(2\pi k \left(\frac{V}{PE}\right) \cos(\theta_l) + 2\pi k \left(\frac{C}{PE}\right)\right) \right. \\ \left. + \sum_{k=1}^{2^n-1} \left[b_k \sin\left(2\pi k \left(\frac{V}{PE}\right) \cos(\theta_l) + 2\pi k \left(\frac{C}{PE}\right)\right) \right] \right] \quad (3.19)$$

Le terme d'erreur de quantification $q(\cdot)$ est négligé car son impact sur la distorsion est faible pour les convertisseurs de haute résolution.

L'équation précédente est assez complexe à manipuler, mais l'utilisation des formules suivantes permettent de simplifier son écriture à l'aide des fonctions de Bessel de première espèce et d'ordre h :

$$\cos[\alpha \cos(p)] = J_0(\alpha) + 2 \sum_{h=1}^{\infty} J_{2h}(\alpha) (-1)^h \cos(2hp) \quad (3.20)$$

$$\sin[\alpha \cos(p)] = 2 \sum_{h=0}^{\infty} J_{2h+1}(\alpha) (-1)^h \cos((2h+1)p) \quad (3.21)$$

Ces fonctions nommées $J_h(\cdot)$, sont largement utilisées dans la résolution d'équations différentielles de second ordre, en particulier dans le domaine de la propagation d'ondes en coordonnées sphériques. Leurs formes s'apparentent à celle d'une sinusoïde dont l'amplitude serait atténuée par la fonction racine carrée. Leur apparition dans les calculs de signaux sinusoïdaux perturbés par une fonction modélisée également par des fonctions trigonométriques semble donc naturelle.

Comme pour la modélisation polynomiale, on cherche à exprimer les paramètres de la fonction traduisant les non-linéarités en fonction du spectre complexe. En reprenant la notation introduite dans les paragraphes précédents, le signal distordu peut s'exprimer comme la somme du signal quantifié idéal et des contributions des harmoniques théoriques S_h^{Th} . Comme précédemment, le convertisseur est considéré sous la forme d'un système non-linéaire mais à invariance temporelle, ce qui implique que les non-linéarités n'introduisent pas de déphasage supplémentaire sur les harmoniques. A partir de ces hypothèses, la distorsion ne génère que des termes de même nature que le signal d'entrée, dans notre cas donc, uniquement des termes en cosinus.

$$s[l] = u[l] + \sum_{0 \leq h} S_h^{Th} \cos(h\theta_l) \quad (3.22)$$

3.3. EXTRACTION À PARTIR DU SPECTRE DE SORTIE DU CAN

En combinant les équations 3.19 et 3.22, on peut mettre en évidence la contribution des non-linéarités vis-à-vis de la distorsion présente dans le spectre du signal de sortie. Ainsi les équations suivantes relient la puissance des harmoniques aux paramètres a_k et b_k représentatifs des non-linéarités :

$$S_{2h}^{Th} = 2(-1)^h \sum_{k=1}^{2^n-1} [a_k \cos(k\alpha_2) + b_k \sin(k\alpha_2)] J_{2h}(k\alpha_1) \quad 1 \leq h \quad (3.23)$$

$$S_{2h+1}^{Th} = 2(-1)^h \sum_{k=1}^{2^n-1} [b_k \cos(k\alpha_2) - a_k \sin(k\alpha_2)] J_{2h+1}(k\alpha_1) \quad 0 \leq h \quad (3.24)$$

avec

$$\alpha_1 = 2\pi \frac{V}{PE} \quad \alpha_2 = 2\pi \frac{C}{PE} \quad (3.25)$$

Cette nouvelle équation traduit bien ce qui nous apparaissait être une évidence : les distorsions sont liées aux non-linéarités (par les coefficients a_k et b_k) et également à l'amplitude (V) et à la composante continue (C) du signal d'entrée. On peut de plus, remarquer que pour $V = 0$, α_1 vaut 0, et donc qu'il n'y a pas de termes de distorsion.

Deux simplifications sont introduites afin de réduire la complexité de la représentation. Tout d'abord, la modélisation en séries de Fourier de la fonction de non-linéarités est limitée à l'ordre K_{max} . Puis, comme pour la modélisation polynomiale, le nombre d'harmoniques extraites du spectre est également limité à la valeur H_{max} . En introduisant ces deux valeurs dans les équations précédentes, la modélisation peut s'écrire comme un système comportant $H_{max}+1$ équations et $2K_{max}+1$ inconnues. La forme la plus adaptée pour représenter ce système est la forme matricielle suivante :

$$\begin{bmatrix} S_0^{Th} \\ \vdots \\ S_{H_{max}}^{Th} \end{bmatrix} = T_{K_{max}} \cdot \begin{bmatrix} a_0 \\ \vdots \\ b_{K_{max}} \end{bmatrix} \quad (3.26)$$

où $T_{K_{max}}$ est une matrice $(H_{max}+1, 2K_{max}+1)$ décrite par :

$$T_{K_{max}} = \begin{bmatrix} \frac{1}{2} & A_1^0 & \dots & A_{K_{max}}^0 & B_1^0 & \dots & B_{K_{max}}^0 \\ 0 & A_1^1 & \dots & A_{K_{max}}^1 & B_1^1 & \dots & B_{K_{max}}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & A_1^{H_{max}} & \dots & A_{K_{max}}^{H_{max}} & B_1^{H_{max}} & \dots & B_{K_{max}}^{H_{max}} \end{bmatrix} \quad (3.27)$$

Pour simplifier les notations, la matrice $T_{K_{max}}$ est décomposée en vecteurs colonnes A_k et B_k dont les expressions sont :

$$A_k = \begin{bmatrix} J_0(k\alpha_1) \cos(k\alpha_2) \\ -2J_1(k\alpha_1) \sin(k\alpha_2) \\ \vdots \\ (-1)^l 2J_{2l}(k\alpha_1) \cos(k\alpha_2) \\ (-1)^{l+1} 2J_{2l+1}(k\alpha_1) \sin(k\alpha_2) \\ \vdots \end{bmatrix} \quad B_k = \begin{bmatrix} J_0(k\alpha_1) \sin(k\alpha_2) \\ -2J_1(k\alpha_1) \cos(k\alpha_2) \\ \vdots \\ (-1)^l 2J_{2l}(k\alpha_1) \sin(k\alpha_2) \\ (-1)^{l+1} 2J_{2l+1}(k\alpha_1) \cos(k\alpha_2) \\ \vdots \end{bmatrix} \quad (3.28)$$

Il est important de signaler que la matrice $T_{K_{max}}$ est définie pour un type d'application, à savoir, pour des valeurs α_1 et α_2 données. Une fois cette matrice calculée, l'effet d'une courbe de non-linéarités précise peut être observée sur le spectre de sortie. Il suffit de déterminer les coefficients a_k et b_k , de procéder à la multiplication matricielle fournie par l'équation 3.26 afin d'estimer la valeur des harmoniques \hat{S}_h . La figure 3.7 illustre le synoptique du calcul.

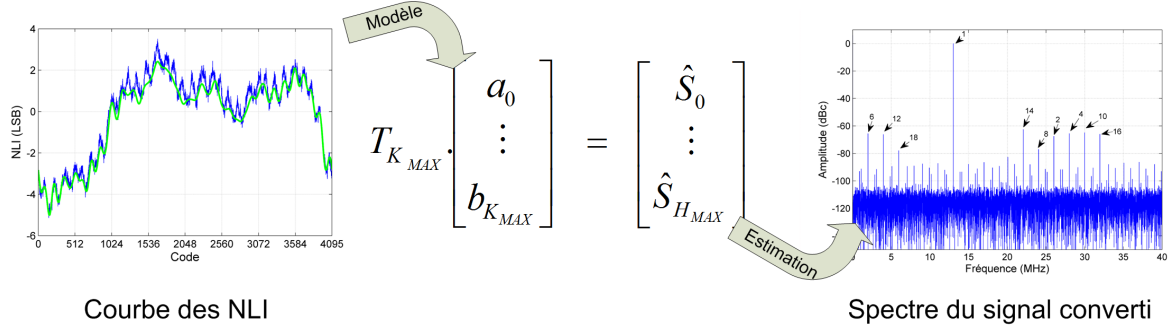


FIG. 3.7 – Estimation des harmoniques à partir du modèle de la courbe des NLI

Optimisation du modèle

La modélisation peut être simplifiée en considérant la plupart des convertisseurs utilisent un signal d'entrée dont la composante continue vaut la moitié de la pleine-échelle ($C = \frac{PE}{2}$). Cette considération implique que la constante α_2 ait pour valeur π . Les équations 3.23 et 3.24 se simplifient et deviennent alors :

$$S_{2h}^{Th} = 2(-1)^h \sum_{k=1}^{2^n-1} a_k (-1)^k J_{2h}(k\alpha_1) \quad 1 \leq h \quad (3.29)$$

$$S_{2h+1}^{Th} = 2(-1)^h \sum_{k=1}^{2^n-1} b_k (-1)^k J_{2h+1}(k\alpha_1) \quad 0 \leq h \quad (3.30)$$

La relation matricielle 3.26 peut alors se décomposer en deux produits matriciels, l'un reliant les harmoniques paires avec les paramètres a_k (3.31), et l'autre reliant les harmoniques impaires avec les paramètres b_k (3.32). Cette simplification permet de passer à une matrice $T_{K_{max}}$ de taille $(H_{MAX}, 2 \times K_{MAX} + 1)$ en deux matrices indépendantes de taille $(H_{MAX}/2, K_{MAX} + 1)$ et $(H_{MAX}/2, K_{MAX})$.

$$\begin{bmatrix} S_0^{Th} \\ S_2^{Th} \\ \vdots \\ S_{H_{max}-1}^{Th} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & A_1^0 & \dots & A_{K_{max}}^0 \\ 0 & A_1^2 & \dots & A_{K_{max}}^2 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & A_1^{H_{max}-1} & \dots & A_{K_{max}}^{H_{max}-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K_{max}} \end{bmatrix} \quad (3.31)$$

$$\begin{bmatrix} S_1^{Th} \\ S_3^{Th} \\ \vdots \\ S_{H_{max}}^{Th} \end{bmatrix} = \begin{bmatrix} B_1^1 & \dots & B_{K_{max}}^1 \\ B_1^3 & \dots & B_{K_{max}}^3 \\ \vdots & \ddots & \vdots \\ B_1^{H_{max}} & \dots & B_{K_{max}}^{H_{max}} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{K_{max}} \end{bmatrix} \quad (3.32)$$

Le terme a_0 représente la valeur moyenne de la courbe de non-linéarités. Il est facilement calculable et n'intervient que dans le calcul de la contribution de la distorsion liée à la fondamentale. Ce terme sera donc mis de côté dans le reste des discussions de cette étude. Grâce à cette simplification, les deux relations matricielles précédentes présentent les mêmes dimensions, à savoir un système linéaire à $H_{MAX}/2$ équations et K_{MAX} inconnues. Comme les paramètres a_k et b_k représentent les projections d'une fonction dans une base orthonormée, ils sont linéairement indépendants. Dans ces conditions,

3.3. EXTRACTION À PARTIR DU SPECTRE DE SORTIE DU CAN

la résolution du système est possible lorsque le nombre d'inconnues égalise le nombre d'équations, soit lorsque $K_{MAX} = H_{MAX}/2$. Pour une valeur de H_{MAX} fixée, augmenter la valeur de K_{MAX} n'apporte aucune information supplémentaire. Cette remarque est vérifiée lorsque l'on résout le système pour différentes valeurs de H_{MAX} et K_{MAX} . La figure 3.8 illustre ce propos. Les deux estimations des courbes des NLI sont identiques, par contre, les calculs mis en oeuvre sont beaucoup plus complexes dans le second cas.

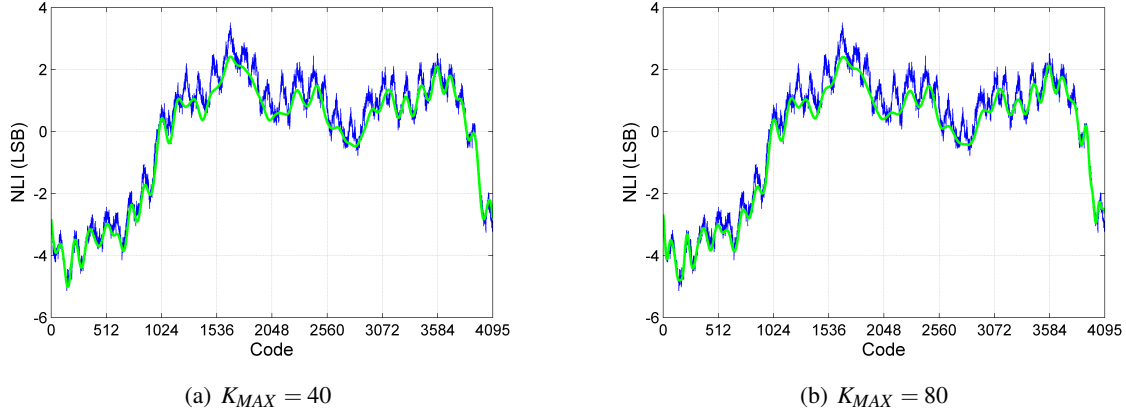


FIG. 3.8 – Estimation de la courbe des NLI pour $H_{MAX} = 80$ en fonction de la valeur de K_{MAX}

Utilisation en vue de l'estimation des non-linéarités

Afin d'obtenir une estimation des paramètres a_k et b_k à partir du nouveau modèle établi, il faut inverser la matrice $T_{K_{max}}$. Pour cela, un algorithme de type SVD (*Singular Value Decomposition*) est utilisé [P⁺92]. Pour une matrice T de taille $(m \times n)$, cet algorithme calcule les éléments suivants :

- les valeurs propres v_p de la matrice T , (au nombre de $\min(m, n)$)
- la matrice U de changement de base de l'espace vectoriel de dimension $(m \times m)$ vers l'espace vectoriel de dimension $(m \times n)$
- la matrice V de changement de base de l'espace de dimension $(n \times n)$ vers l'espace vectoriel de dimension $(m \times n)$.

La matrice T peut alors s'écrire sous la forme $T = U.S.^t.V$, où S est la matrice de taille $(m \times n)$ dont tous les éléments sont nuls sauf les éléments diagonaux qui sont égaux aux valeurs propres v_p .

La matrice pseudo-inverse de T est alors donnée par la relation $T^{-1} = V.S^+.^t.U$. La matrice S^+ est une matrice de taille $(n \times m)$ ayant pour éléments diagonaux les valeurs inverses des valeurs propres. Pour des questions de stabilité, ces valeurs inverses ne sont calculées que pour des valeurs de v_p suffisamment différentes de zéro. La méthode utilisée remplace par 0 les valeurs en $1/v_p$ dépassant la valeur 2 [P⁺92]. Ce choix, pour le moment purement empirique, fait l'objet de futurs travaux.

A partir de l'inverse de la matrice $T_{K_{max}}$ et d'une estimation de la puissance des harmoniques \hat{S}_h effectuée sur le spectre du signal converti, le calcul matriciel 3.33 permet d'obtenir les valeurs des paramètres a_k et b_k . Les calculs sont bien évidemment simplifiés en utilisant les relations matricielles fournies par les

équations 3.31 et 3.32.

$$\begin{bmatrix} a_0 \\ \vdots \\ b_{K_{max}} \end{bmatrix} = T_{K_{max}}^{-1} \cdot \begin{bmatrix} \hat{S}_0 \\ \vdots \\ \hat{S}_{H_{max}} \end{bmatrix} \quad (3.33)$$

Cette implémentation a été réalisée sous MATLAB. Les figures 3.9 représentent l'estimation de la courbe des NLI pour différentes valeurs de K_{MAX} et H_{MAX} . Le fait d'augmenter ces valeurs permet de mieux suivre la variation de la courbe, mais il faut garder à l'esprit qu'utiliser 1000 harmoniques dans nos calculs revient majoritairement à prendre le bruit en considération. Un moyennage de ce bruit doit être effectué afin d'éliminer les composantes aléatoires et de ne conserver que les phénomènes dus à la distorsion, aussi minimes qu'ils puissent être.

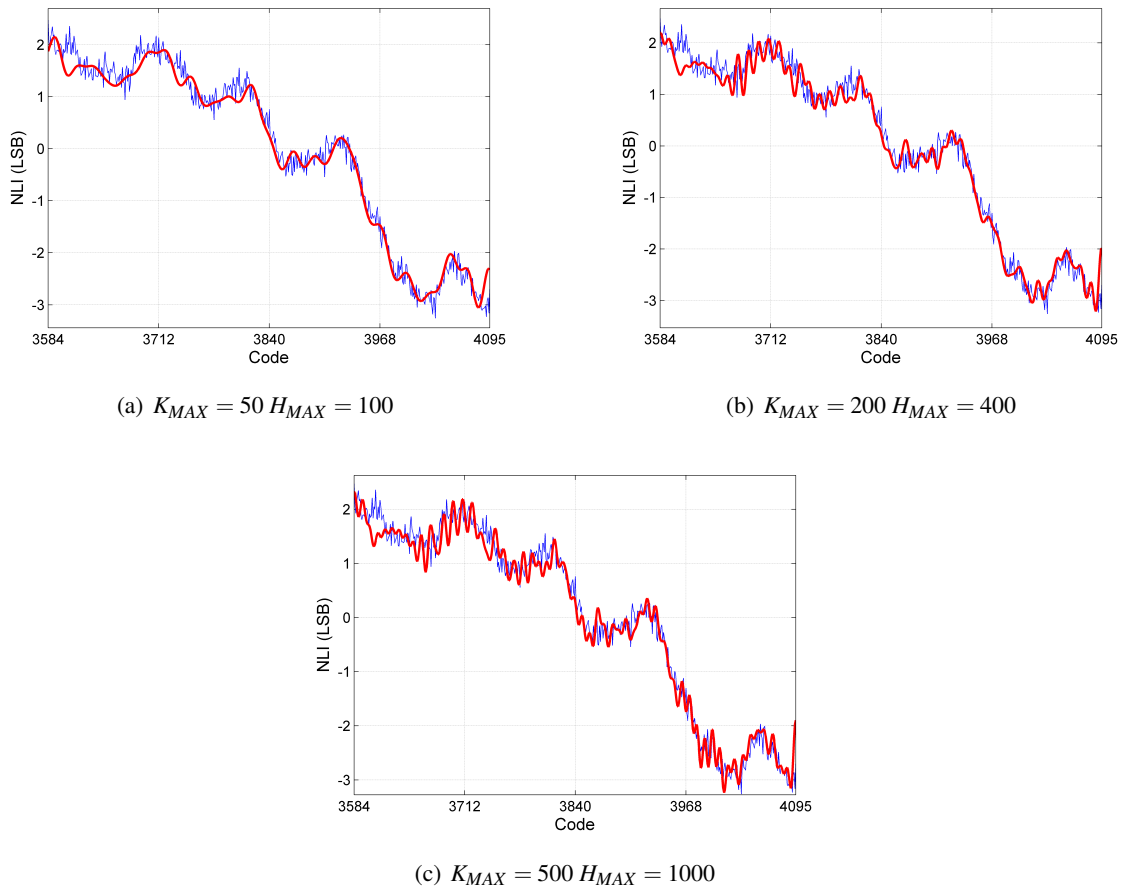


FIG. 3.9 – Estimation de la courbe des NLI en fonction de K_{MAX} et H_{MAX}

3.4 Conclusion

Dans ce chapitre, trois méthodes d'extraction des non-linéarités ont été exposées. La première méthode utilise une approche statistique qui requiert un nombre de points très élevé pour obtenir une estimation correcte de la courbe des NLI. Les deux autres méthodes utilisent une approche analytique basée sur

3.4. CONCLUSION

l'exploitation fréquentielle du signal. A partir du spectre complexe du signal converti, et d'une modélisation des non-linéarités (Chebyshev ou Fourier), les paramètres du modèle peuvent être retrouvés. La première modélisation, basée sur des polynômes, ne permet pas la retranscription des variations brusques de la courbe de NLI. Notre nouvelle méthode, basée sur une décomposition en série de Fourier, est plus à même de reproduire ces variations de par la nature trigonométrique des fonctions qui la composent.

Les deux dernières méthodes réclament un nombre de points très inférieur à celui requis pour la méthode par histogramme (environ 16 fois moins), mais elles sont également plus complexes à mettre en oeuvre d'un point de vue mathématiques. Dans le cadre d'une compensation *offline*, les trois méthodes peuvent être considérées selon la nature de la correction que l'on souhaite apporter. Par contre, dans le cadre d'une compensation embarquée, l'étude de l'implémentation physique est nécessaire afin de conclure sur le choix d'une de ces méthodes.

Chapitre 4

Etude de l'efficacité de la compensation *offline*

Le chapitre 4 traite de la mise en place de la compensation à partir d'un convertisseur conçu spécialement pour être utilisé dans le cadre de cette thèse. Nous avons mis en place une carte de démonstration sur laquelle ce nouveau convertisseur est associé à un composant reprogrammable qui intègre une table de correction. Ce véhicule de test nous permet d'évaluer la robustesse de la méthode de compensation en fonction de l'application.

4.1 Introduction

Au cours des deux chapitres précédents, nous avons exposé notre intérêt pour une compensation de type Look Up Table statique à une dimension et nous avons également exposé les différentes techniques permettant d'estimer les non-linéarités nécessaires au remplissage des tables de correspondance. Ce calcul s'effectue dans ce chapitre de manière extérieure au composant, par exemple par les ressources de calcul d'un testeur durant la production du produit, c'est donc une calibration dite *offline*. La mise en oeuvre de la technique au travers d'un véhicule de test complet est exposé ci dessous.

La première section détaille le convertisseur et le composant reprogrammable de type FPGA utilisés pour cette étude. Ensuite, une section complète est dédiée aux mesures de robustesse de la compensation effectuée sur notre démonstrateur. Cette partie met en concurrence les différentes méthodes de remplissage de la table, et met en avant la robustesse et la sensibilité de la méthode corrective vis à vis de divers paramètres telle que la fréquence d'échantillonnage ou la fréquence d'entrée. Finalement, la dernière section conclut sur l'efficacité de la compensation *offline*.

4.2 Véhicule de Test

Dans le chapitre 1, l'architecture à repliement et interpolation a été détaillée afin de sensibiliser le lecteur aux difficultés de la conception de convertisseurs haute vitesse. Lors de cette phase, le concepteur doit

4.2. VÉHICULE DE TEST

constamment faire un compromis entre la linéarité et la vitesse d'exécution du bloc qu'il met en oeuvre au sein de l'architecture. Généralement, il privilégie la linéarité de la fonction afin d'obtenir une conversion correcte, et ceci, au détriment de la vitesse globale du composant. Pour cette étude, un convertisseur spécifique a été conçu en inversant les priorités : la vitesse d'exécution est préférée à la linéarité optimale du bloc. L'idée ici est donc d'utiliser les techniques de compensation pour améliorer, après fabrication, la non-linéarité du convertisseur. Dans ce contexte une équipe de conception de NXP a réalisé un CAN très performant en terme de rapport signal sur bruit à haute vitesse mais avec présentant des non-linéarités importantes. Les caractéristiques de ce nouveau convertisseur sont exposés dans les paragraphes suivants. Tout au long de ce chapitre, ce nouveau convertisseur sera référencé sous le nom d' AD_{DEMO} .

Pour effectuer nos diverses implémentations de la méthode de compensation, un composant reprogrammable est utilisé. En effet, étant donné que la post-correction intervient entièrement dans le domaine numérique, l'utilisation de composant de type FPGA nous permet un grand degré de souplesse dans la mise en oeuvre de ces méthodes. Son principe de fonctionnement et sa programmation sont exposés dans les derniers paragraphes de cette section. La figure 4.1 illustre la structure du démonstrateur : une carte dédiée au convertisseur, et une carte pour la compensation et l'acquisition des données.

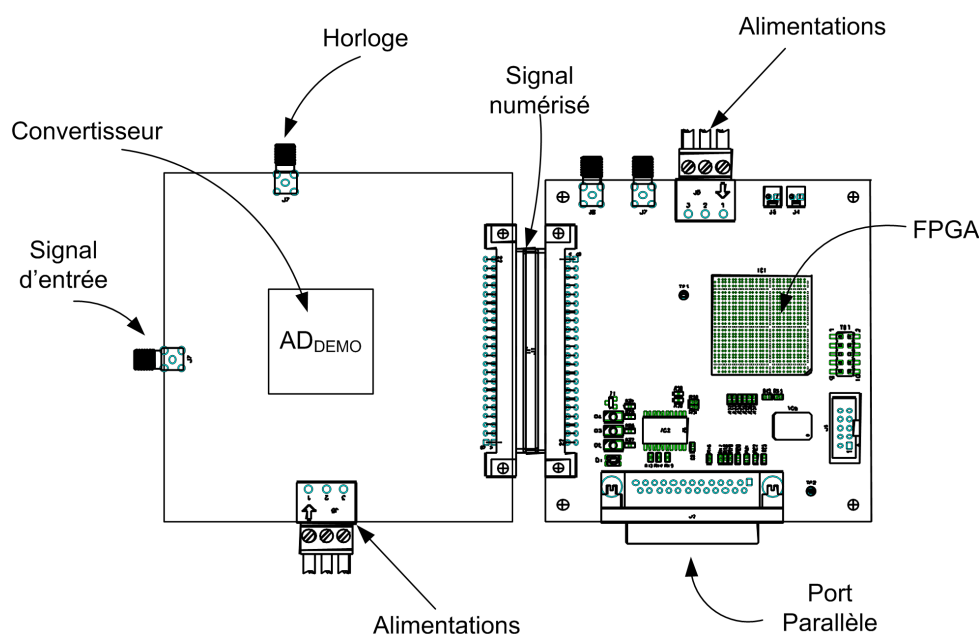


FIG. 4.1 – Cartes de démonstration

4.2.1 Le convertisseur AD_{DEMO}

Le convertisseur AD_{DEMO} est une déclinaison d'un convertisseur commercial issu du port-folio de NXP Semiconductors. Le convertisseur originel est de type à repliement et interpolation ; il possède une résolution de 12bits et une fréquence d'échantillonnage moyenne de 80MHz. La relaxation des contraintes sur l' AD_{DEMO} est effectué au niveau de l'appariement des capacités et des transistors.

Dans le chapitre 1, la problématique linéarité/vitesse liée aux transistors a été exposée très succinctement.

Cependant, dans le cadre de cette étude, elle sera amplement suffisante pour comprendre que l'augmentation de la vitesse du convertisseur implique l'apparition des non-linéarités. La conception de l' AD_{DEMO} a donc suivi cette logique. L'équipe conception a relâché les contraintes de non-linéarités du composant de base afin d'augmenter sa vitesse d'exécution tout en conservant un bruit de quantification au même niveau que le composant originel. En effet, la post-correction peut agir sur la distorsion du signal, mais pas sur la quantification : la correction intervenant dans le monde numérique, la plus petite correction qui peut être appliquée est de l'ordre du LSB. Les paragraphes suivants exposent les caractéristiques de l' AD_{DEMO}

Caractéristiques du convertisseur AD_{DEMO}

Afin de présenter les caractéristiques du convertisseur AD_{DEMO} , une étude comparative est effectuée vis à vis du convertisseur commercial qui sera dénommé ici tout simplement par le terme AD . L'architecture est donc du type à repliement et interpolation et la résolution du convertisseur est de 12 bits. La modification de la conception interne ne change rien à ces éléments. La comparaison des performances des deux convertisseurs se base donc sur les paramètres statiques et dynamiques cités dans le premier chapitre.

En observant la courbe des non linéarités (figure 4.2), il apparaît clairement que le convertisseur AD_{DEMO} possède beaucoup plus de non linéarités que le convertisseur originel. Ces non linéarités impactent directement les performances dynamiques, comme on peut l'observer sur les spectres respectifs des deux convertisseurs (figure 4.3).

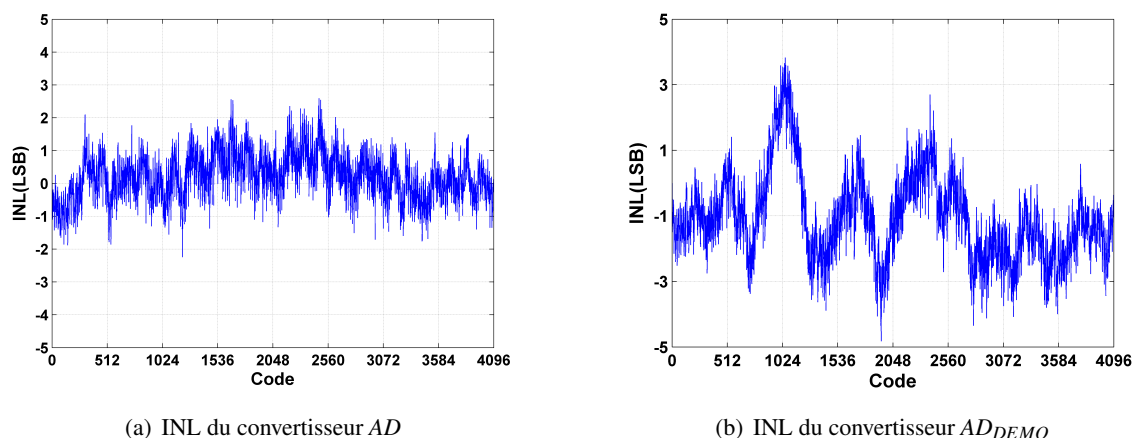


FIG. 4.2 – Comparaison des NLI

Les courbes de NLI sont calculées pour une fréquence d'échantillonnage de 80MHz et une fréquence d'entrée de 4.43MHz. La méthode de mesure est celle de l'histogramme. Les spectres de puissance sont obtenus pour la même fréquence d'échantillonnage, mais pour une fréquence d'entrée de 93MHz. Bien évidemment, la relation de cohérence est respectée à chaque fois.

Afin de vérifier le comportement des convertisseurs vis à vis de la fréquence d'échantillonnage, le paramètre $SFDR$ a été mesuré pour diverses fréquences d'horloge. Ces mesures ont été effectuées sur une population de 5 convertisseurs AD et de 5 convertisseurs AD_{DEMO} afin d'obtenir une valeur moyenne de

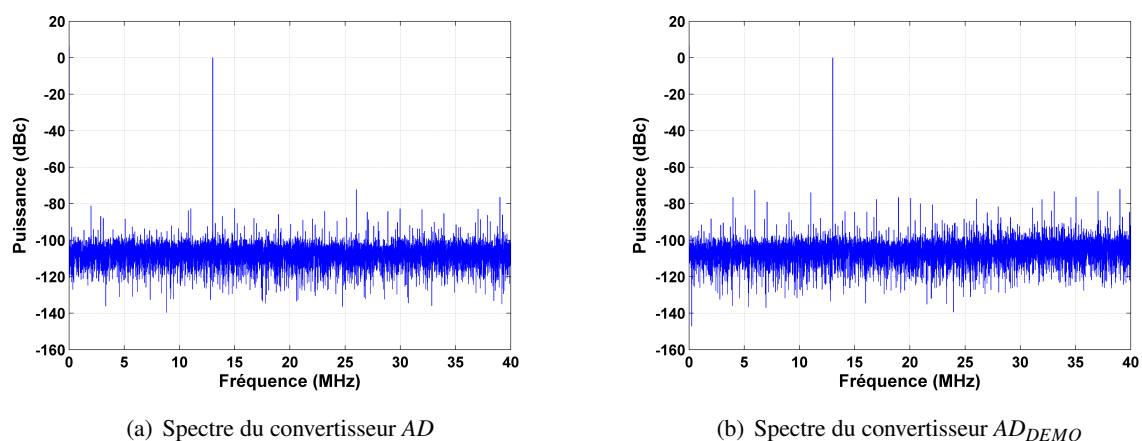


FIG. 4.3 – Comparaison des Spectres

ce paramètre. Le résultat est présenté sur la figure 4.4. Comme vu précédemment, le convertisseur AD_{DEMO} possède un $SFDR$ moins bon que le convertisseur AD , mais il conserve sa valeur au fur et à mesure que la fréquence d'échantillonnage augmente, tandis que le convertisseur AD voit sa valeur diminuer de façon considérable au fur et à mesure de cette augmentation. La conception du convertisseur AD_{DEMO} répond donc bien aux attentes de ses concepteurs, et aux besoins de notre étude.

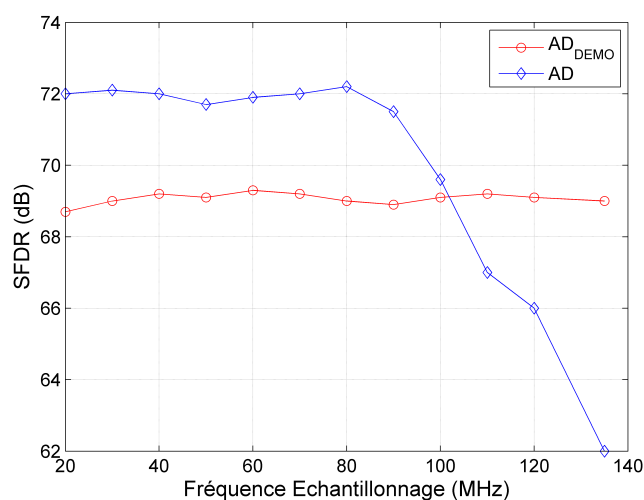


FIG. 4.4 – Comparaison de la linéarité des convertisseurs en fonction de la fréquence d'échantillonnage

4.2.2 Spécifications et configuration du FPGA

Maintenant que nous possédons un convertisseur adapté à nos besoins, il nous faut mettre en oeuvre la partie correction. Étant donné que la correction s'effectue numériquement, l'utilisation d'un composant numérique reprogrammable paraît évidente. En effet, puisque l'on veut tester diverses implémentations de correction sans passer par une reconception complète et coûteuse du convertisseur, cette solution nous

semble la plus adaptée.

Les ressources nécessaires à implanter dans le FPGA pour la compensation *offline* ne sont pas très importantes. Une simple table de 2^{12} valeurs codée sur 12bits est nécessaire, et une machine d'état permettant le contrôle du remplissage de ces valeurs. L'ensemble des FPGA présent actuellement sur le marché surpasse de loin ces prérequis. Dans cette étude, nous avons utilisé un FPGA d'Altera : le Stratix II [Alt05]. Un circuit a été synthétisé avec le logiciel Altera Quartus II, puis implanté sur le FPGA cible. Celui-ci permet à la fois de stocker les valeurs du signal converti au sein d'une mémoire de type FIFO, et également de recevoir les valeurs de correction à appliquer au signal converti. L'acquisition/configuration s'effectue via le port parallèle d'un PC. Un programme réalisé sous LabVIEW permet de contrôler les différents processus et également d'extraire les non-linéarités.

La fréquence maximale des échantillons en entrée pour ce circuit est de 266MHz, celle-ci est largement supérieure aux fréquences d'échantillonnage du convertisseur.

Chronologie de l'expérience

L'opération de compensation se déroule en 3 étapes :

- Une acquisition des données est effectuée au travers du FPGA comme indiqué sur la figure 4.5. Celui-ci stocke dans une mémoire de type FIFO les données issues du convertisseur AD_{DEMO} . L'acquisition est effectuée en respectant la relation de cohérence du système (cf chap1). A partir de ces données, l'ordinateur effectue le calcul de la courbe des non linéarités selon la méthode choisie par l'utilisateur (histogramme, spectrale). Les mesures dynamiques calculées à partir du spectre sont également affichées et stockées pour pouvoir être comparées par la suite.

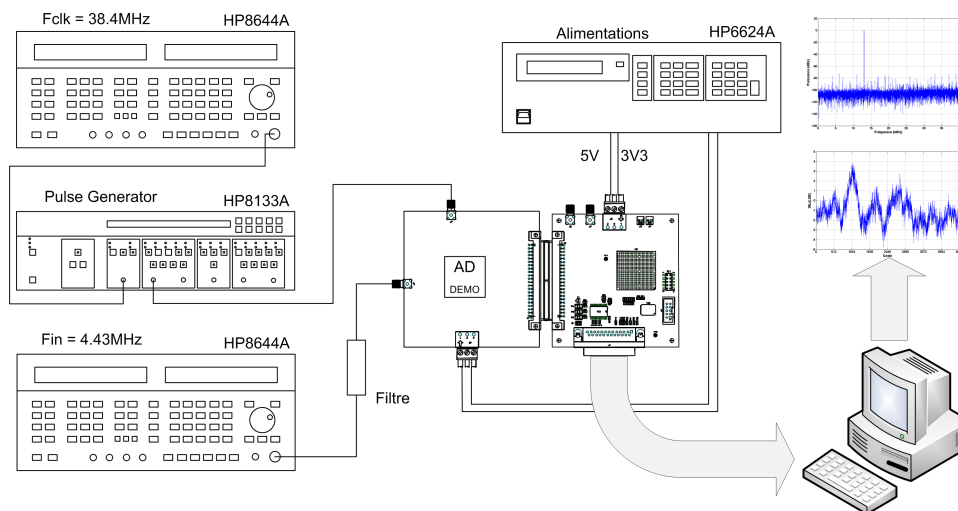


FIG. 4.5 – Acquisition et calcul de la courbe des non-linéarités

- A partir du calcul de la courbe des non linéarités, l'ordinateur écrit via le port parallèle les valeurs de corrections de la table de correspondance du FPGA. La figure 4.6 illustre ce principe.
- Une fois la table de correspondance remplie, une seconde acquisition est effectuée. Le FPGA procède à la substitution des codes de sortie en fonction de la correction à apporter, et stocke dans la mémoire FIFO les valeurs corrigées. L'ordinateur peut alors récupérer ces données pour effectuer les mesures de

4.3. ETUDE DE LA ROBUSTESSE DE LA MÉTHODE DE COMPENSATION

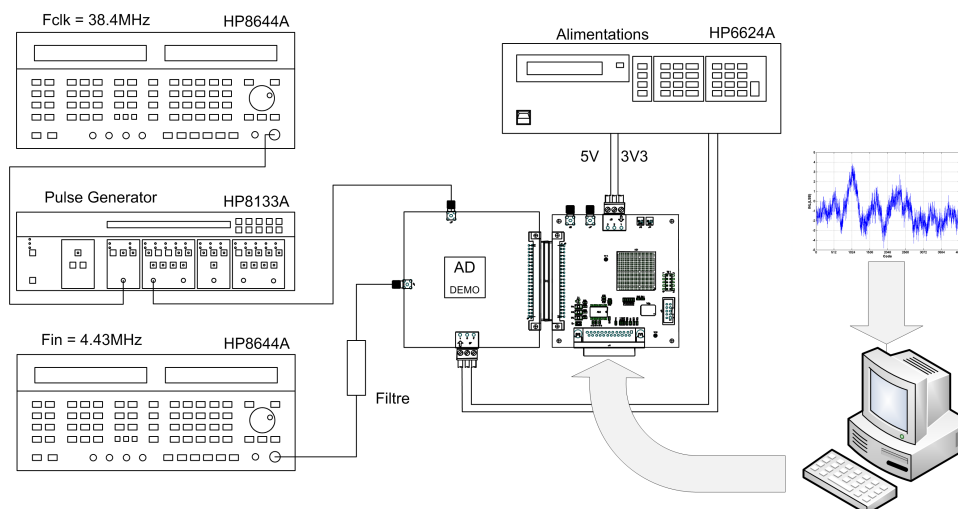


FIG. 4.6 – Remplissage de la table de correspondance

paramètres statiques et dynamiques. La figure 4.7 illustre cette phase. Les valeurs des paramètres après correction sont comparées aux valeurs stockées lors de la phase 1, et l'efficacité de la compensation est alors établie en terme de gain en dB ou en nombre de bits effectifs. Cette efficacité sera exposée plus en détails dans la section suivante.

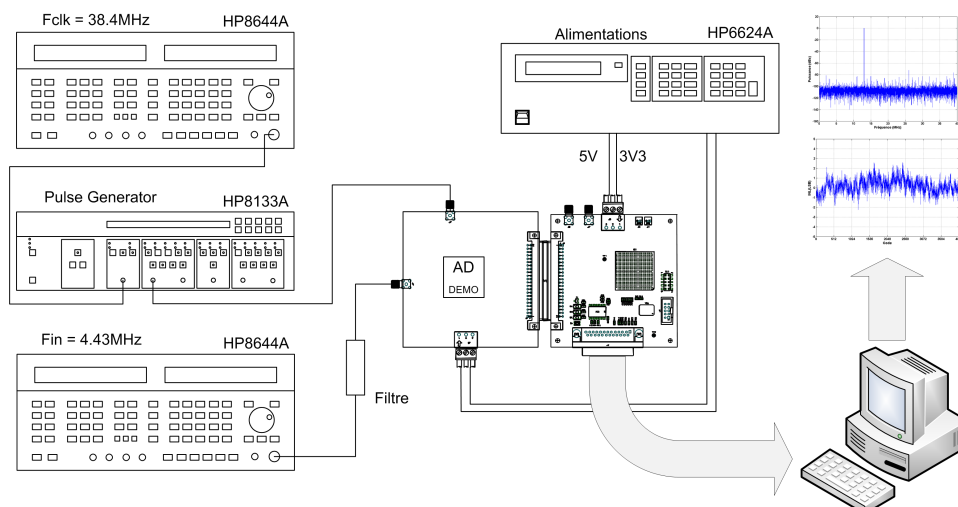


FIG. 4.7 – Acquisition des données corrigées

A partir de ce démonstrateur, nous sommes désormais capables d'effectuer une étude approfondie de la stabilité et de la robustesse de la méthode de compensation. La suite de ce chapitre est dédiée à cette étude.

4.3 Etude de la robustesse de la méthode de compensation

Dans ce chapitre, la compensation s'effectue de façon *offline*, ce qui ne nous impose aucune contrainte sur le choix de la technique de calcul des non-linéarités. La méthode par histogramme peut donc être

appliquée à partir d'un très grand nombre de points, et la méthode spectrale peut également être utilisée avec une valeur très grande pour le paramètre H_{max} . Étant donné que l'histogramme constitue la valeur de référence du calcul des non-linéarités, nous établirons tout d'abord une comparaison entre les effets des deux compensations. L'efficacité de la compensation est ensuite testée selon la valeur des fréquences d'entrée et d'échantillonnage utilisées. Finalement, la compensation est évaluée pour une valeur de table de correspondance établie en fonction de la fréquence d'entrée.

4.3.1 Comparaison des méthodes de calcul de la NLI

Dans le chapitre 3, différentes méthodes de calculs d'extraction des non-linéarités ont été exposées. La méthode par histogramme constitue la méthode de référence pour l'obtention de la courbe des non-linéarités. Cependant, les méthodes spectrales permettent également d'obtenir une allure suffisamment précise de cette même courbe. Étant donné que la table de correction est constituée de valeurs de type *entier*, l'information inférieure au LSB ne peut être significative pour la compensation. Ainsi la précision obtenue par l'histogramme n'est pas forcément utile à l'obtention d'une meilleure table de correction. La figure 4.8 met en évidence la différence de précision des deux méthodes : le calcul par la méthode spectrale ne fournit qu'une allure générale de la courbe de NLI, alors que le calcul par histogramme fournit une valeur beaucoup plus précise de la courbe de NLI. La figure 4.9 montre que la fonction arrondie rapproche la valeur des deux estimations, même si la représentation effectuée à partir du spectre constitue toujours une version arrondie et moyennée de la courbe.

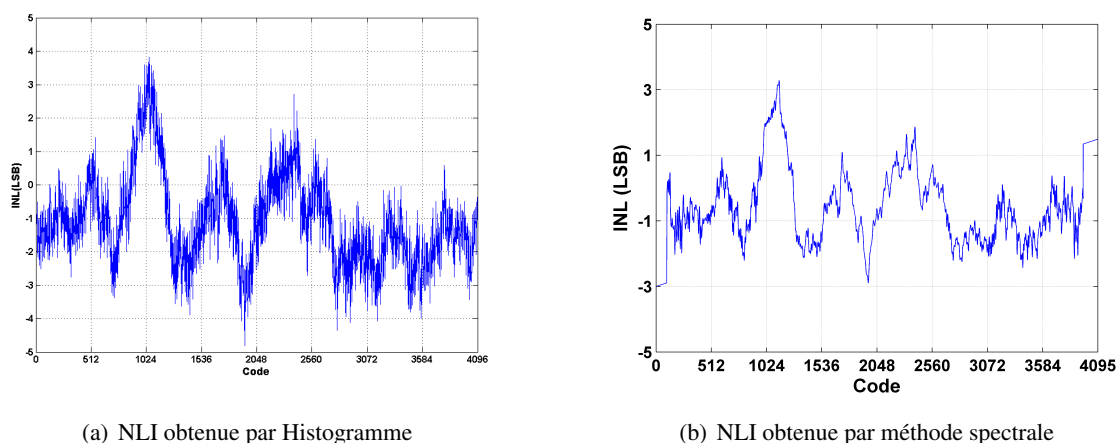


FIG. 4.8 – Comparaison des courbes de NLI selon la méthode de calcul

L'observation des effets de ces deux compensations a donc été menée afin de vérifier l'impact des erreurs d'estimation. Les spectres résultants des deux corrections apparaissent sur la figure 4.10, et les valeurs caractéristiques figurent dans le tableau 4.1. Les résultats apparaissent surprenants : la meilleure correction est obtenue non pas à partir de l'estimation effectuée par histogramme, mais par l'estimation basée sur l'analyse spectrale. Ce paradoxe fait l'objet de futurs travaux, mais l'effet lissage obtenu par l'analyse spectrale, semble être plus adapté pour la compensation, même si la différence est peu significative.

La méthode de calcul n'influence donc que très peu la performance de la correction *offline*. La différence

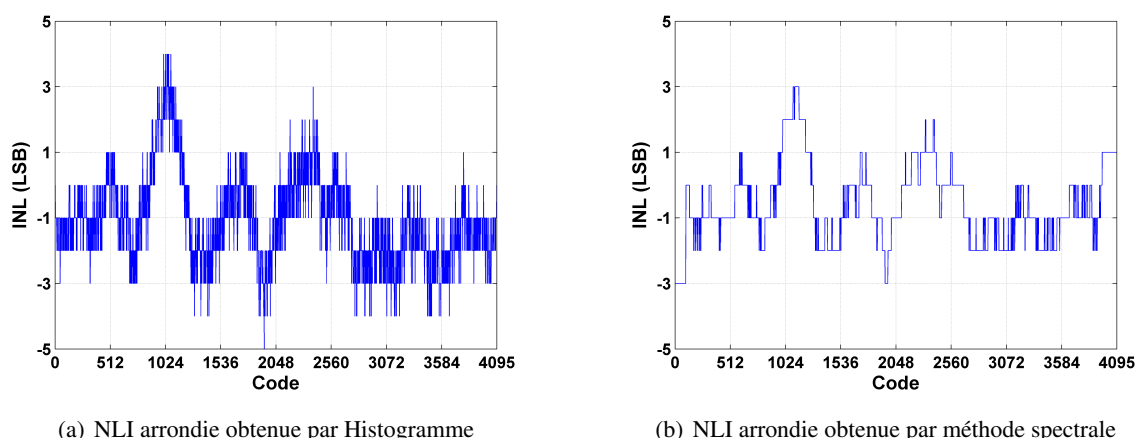


FIG. 4.9 – Comparaison des courbes de NLI arrondies selon la méthode de calcul

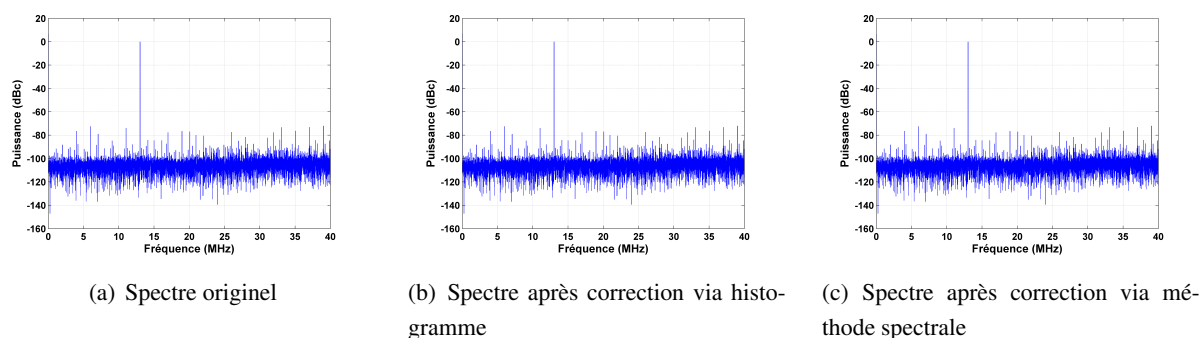


FIG. 4.10 – Comparaison des spectres avant et après corrections selon la méthode de calcul

Param.	Sans Corr.	Corr. Histo	Corr. Spectr.
SNR (dB)	61.07	65.37	66.53
SFDR (dB)	66.59	84.44	84.90
THD (dB)	-74.44	-82.62	-83.42
SINAD (dB)	60.87	65.28	66.88
ENOB (bits)	9.82	10.58	10.72

TAB. 4.1 – Amélioration des paramètres dynamiques en fonction de la méthode de calcul

majeure des deux méthodes intervient vis à vis du nombre de points nécessaires à l'obtention des valeurs. Nous allons maintenant observer l'effet de la correction suivant les paramètres fréquentiels.

4.3.2 Rappels sur l'échantillonnage et le repliement spectral

Avant de nous attarder sur l'étude fréquentielle de l'efficacité de la compensation, nous procédons à des rappels rapides relatifs à l'échantillonnage et au repliement spectral.

Théorème de Nyquist-Shannon

Le théorème de Nyquist-Shannon énonce que la fréquence d'échantillonnage f_s d'un signal doit être égale ou supérieure au double de la fréquence maximale contenue dans ce signal, afin de convertir ce signal d'une forme analogique à une forme numérique. La figure 4.11 illustre le cas parfait où le spectre du signal analogique (représenté par les pointillés orange) est inclus dans la fenêtre d'observation fréquentielle délimitée par les fréquences $[-\frac{f_s}{2} .. \frac{f_s}{2}]$. D'après la théorie de l'échantillonnage, les images du spectre du signal échantillonné se retrouvent aux multiples des fréquences f_s . En respectant la contrainte fréquentielle énoncée par le théorème de Nyquist-Shannon, on est sûr que le signal observé correspond bien au signal originel.

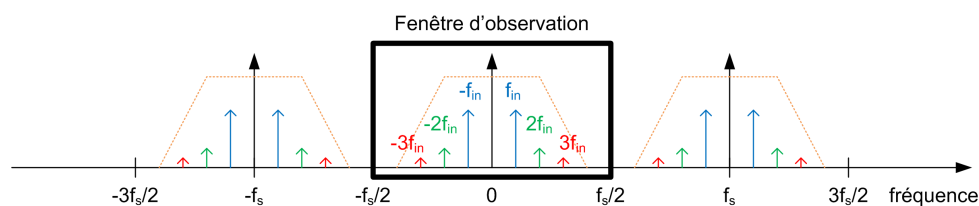


FIG. 4.11 – Spectre d'un signal numérisé en respectant la contrainte de Nyquist-Shannon

Dans le cas où cette contrainte n'est pas respectée, une partie des images supérieures et inférieures du spectre du signal, apparaissent dans la fenêtre d'observation. La figure 4.12 illustre ce propos. La 3ème harmonique ($3f_{in}$) se situe en dehors de la fenêtre d'observation, tandis que les images supérieures et inférieures de la troisième harmonique rentrent dans cette même fenêtre.

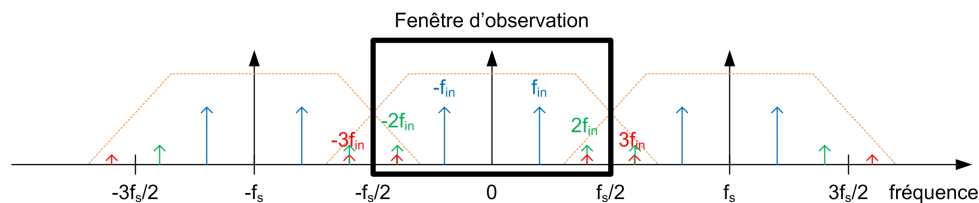


FIG. 4.12 – Spectre d'un signal numérisé ne respectant pas la contrainte de Nyquist-Shannon

De façon générale, le cas idéal n'est que rarement réalisable, car les systèmes d'échantillonnage introduisent des distorsions sur le signal d'entrée qui génèrent des fréquences multiples de la fréquence d'entrée. Ainsi l'échantillonnage d'un signal de fréquence $f_{in} = 30\text{MHz}$ effectué à la fréquence $f_s = 80\text{MHz}$ peut générer l'apparition d'une harmonique d'ordre 2 de fréquence $2 \times f_{in} = 60\text{MHz}$. Cette fréquence ne respecte pas la contrainte du théorème de Nyquist-Shannon, les images supérieures et inférieures du spectre apparaissent donc dans la fenêtre d'observation. Ce phénomène se nomme "repliement spectral", et est illustré ci-dessous.

Repliement spectral

Le repliement spectral s'observe pour tout signal ne respectant pas la contrainte du théorème de Nyquist-Shannon. La zone fréquentielle délimitée par les fréquences $[0.. \frac{f_s}{2}]$ se nomme "zone de Nyquist 1". Elle est représentée par le symbole (1) sur la figure 4.13. Toute fréquence appartenant à cette zone apparaît de façon directe dans la zone d'observation. De façon similaire, une zone (-1) existe pour les fréquences négatives.

La zone fréquentielle délimitée par les fréquences $[\frac{f_s}{2}..f_s]$ se nomme "zone de Nyquist 2", représentée par le symbole (2). Toute fréquence appartenant à cette zone se retrouve en dehors de la fenêtre d'observation, mais à cause de la présence de l'image supérieure, l'image de cette fréquence se retrouve quand même dans la fenêtre. La position de cette fréquence image au sein de la fenêtre d'observation est fournie par une symétrie par rapport à la fréquence $\frac{f_s}{2}$. Cette symétrie traduit le "repliement du spectre".

La figure 4.13 schématise le repliement des différentes zones ((1)(2)(3)(4)) au sein de la fenêtre d'observation.

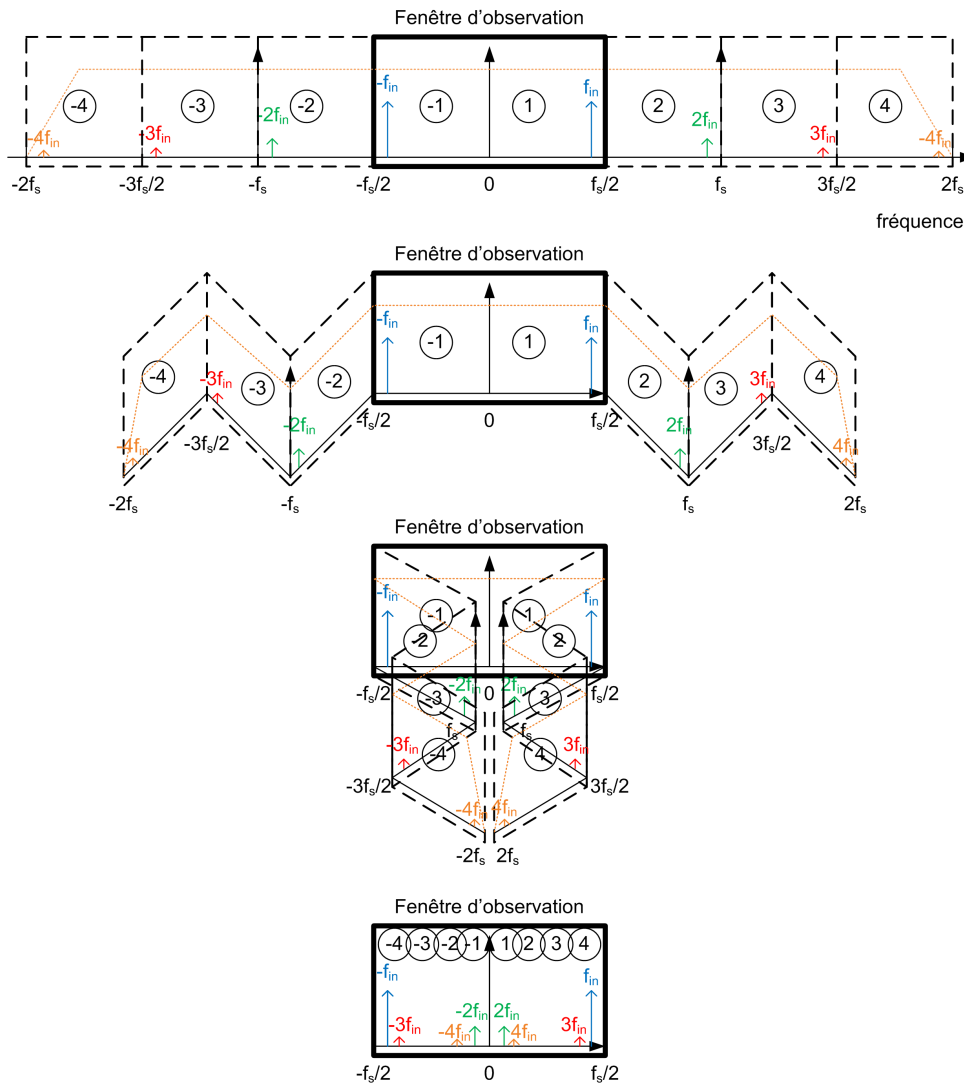


FIG. 4.13 – Principe du repliement spectral

Ce principe de repliement est souvent utilisé en télécommunications afin de retrouver un signal en bande de base. La figure 4.14 illustre un signal analogique à bande spectrale étroite centrée sur une porteuse de fréquence f_{in} . L'échantillonnage de ce signal à une fréquence f_s inférieure à f_{in} permet, grâce au repliement spectral, de retrouver la bande utile au sein de la fenêtre d'observation. Pour réaliser cette opération, il faut que l'étage d'entrée du convertisseur soit robuste aux brusques variations impliquées par la fréquence élevée f_{in} . Si le convertisseur n'est pas suffisamment linéaire à ces fréquences, alors les harmoniques du signal contenu dans la bande utile seront générées et, du fait du repliement, risquent de se retrouver dans la fenêtre d'observation, perturbant alors la reconstruction du signal numérique.

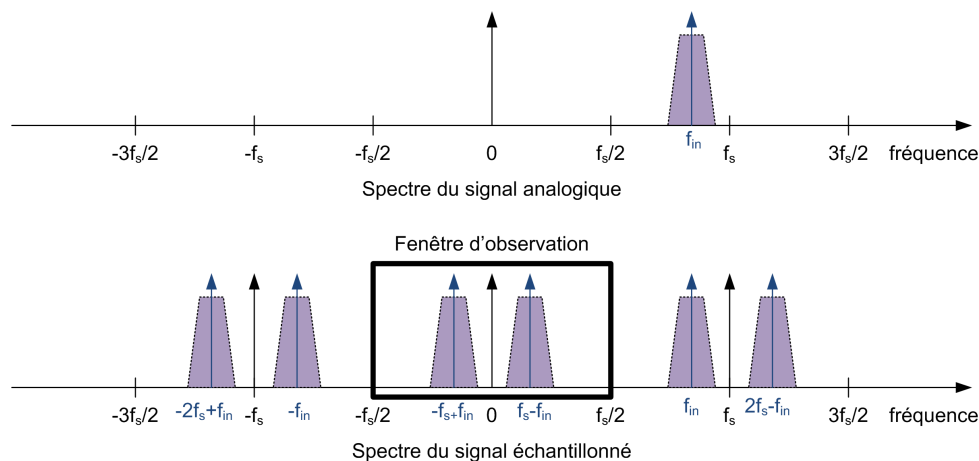


FIG. 4.14 – Utilisation du repliement spectral

Ces rappels nous permettent de justifier l'étude du convertisseur pour des fréquences f_{in} supérieures à la moitié de la fréquence d'échantillonnage f_s .

4.3.3 Efficacité de la compensation selon la variation des fréquences d'utilisation

Précédemment, la relation entre les non-linéarités et la fréquence d'échantillonnage a été démontrée. Il apparaît donc naturel que les non-linéarités varient en fonction du domaine d'applications choisi par l'utilisateur. Afin de vérifier si notre méthode de compensation peut couvrir tout ces domaines, nous procédons à la variation de la fréquence d'entrée, puis à celle de la fréquence d'échantillonnage. En effet, s'il apparaît évident que le comportement des composants internes du convertisseur soit lié intrinsèquement à la fréquence d'échantillonnage ; la dépendance face aux variations de la pente des signaux d'entrée doit également être pris en considération. Ainsi, la première partie de cette étude traite du comportement de la compensation selon la fréquence d'entrée, et la seconde partie, du comportement de la compensation selon la fréquence d'échantillonnage.

Efficacité de la compensation en fonction de la fréquence du signal d'entrée

Le convertisseur originel utilisé pour la déclinaison de l' AD_{DEMO} est destiné à couvrir la plage de fréquence d'entrée comprise entre quelques MégaHertz et 175MHz. Notre plage de test sera donc identique

4.3. ETUDE DE LA ROBUSTESSE DE LA MÉTHODE DE COMPENSATION

et la fréquence d'échantillonnage sera fixée à 80MHz.

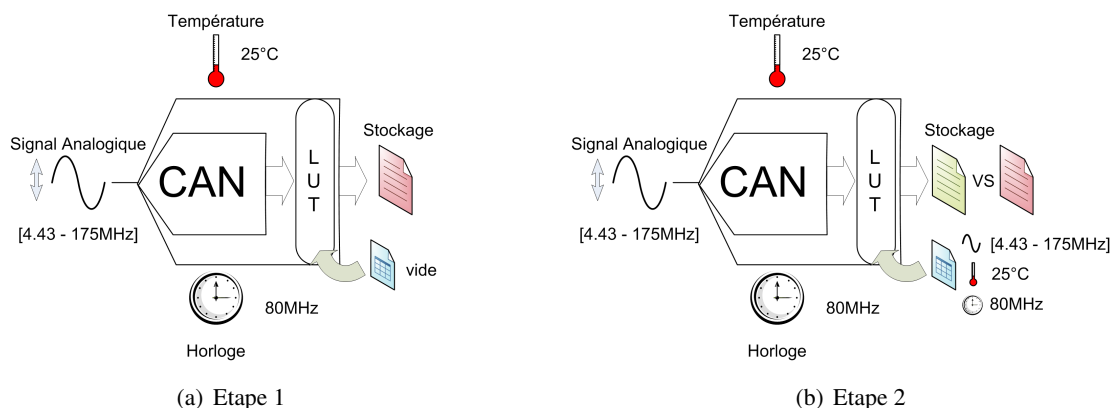


FIG. 4.15 – Compensation avec la LUT calculée en fonction de la fréquence d'entrée

Le déroulement de la mesure est la suivante (figure 4.15) :

- Une fréquence d'entrée est choisie entre 4.43MHz et 175MHz
- Une acquisition est lancée et les paramètres dynamiques sont stockés
- La courbe de non-linéarité est mesurée pour la fréquence choisie (prise en compte des non-linéarités dynamiques)
- La table est chargée dans le FPGA
- Une seconde acquisition est lancée pour la même fréquence d'entrée et les nouveaux paramètres dynamiques sont stockés.
- L'expérience est réitérée pour une autre fréquence d'entrée.

Pour l'ensemble des expérimentations de ce chapitre, les mesures sont effectuées plusieurs fois sur une population de quelques unités de convertisseurs. Pour chaque convertisseur, les paramètres dynamiques *THD*, *SFDR* et *SINAD* sont observés. Ces paramètres sont enregistrés avant et après correction et sont consultables au travers des figures 4.16.

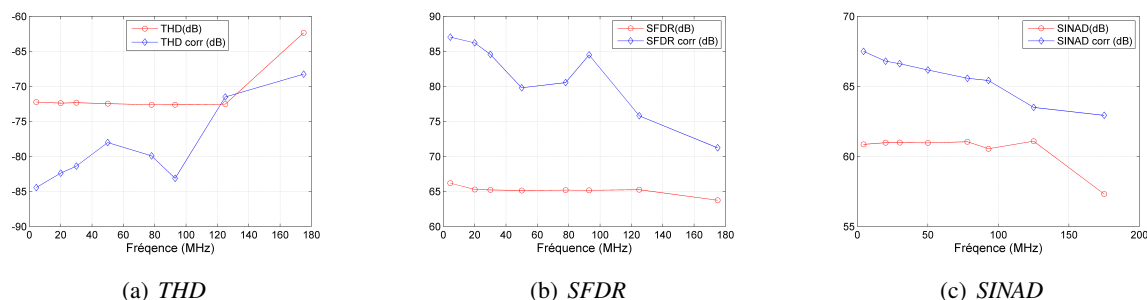


FIG. 4.16 – Évolution de la *THD*, du *SFDR* et du *SINAD* pour une LUT remplie à chaque changement de fréquence d'entrée

La compensation apparaît bien effective et prend en compte les variations des non-linéarités aux différentes fréquences d'entrée. Les courbes rouges correspondent aux valeurs mesurées avant compensation, et les courbes bleues aux valeurs après compensation. L'atténuation du niveau des énergies des harmo-

riques est visible au travers des améliorations de la *THD*, du *SFDR* et du *SINAD*. L'amélioration n'est pas constante sur la gamme de fréquence étudiée. La compensation effectuée pour des fréquences appartenant à la seconde zone de Nyquist possède une amélioration plus faible que pour les fréquences appartenant à la première zone, tandis que la compensation pour les fréquences de la troisième zone tend à se rapprocher des valeurs non corrigées.

La conclusion que l'on peut énoncer ici, est que la correction calculée pour une fréquence d'entrée donnée, améliore bien les performances du convertisseur pour cette même fréquence. Cependant, ce cas d'utilisation est rare dans la pratique. Les applications du type GSM, par exemple, réclament de pouvoir fonctionner sur plusieurs bandes de fréquences, ce qui impose que la correction du convertisseur doit pouvoir s'effectuer indépendamment du choix de la fréquence de calcul de la table. La section suivante tente de répondre à cette attente.

Efficacité de la compensation pour une table calculée à une fréquence d'entrée fixe

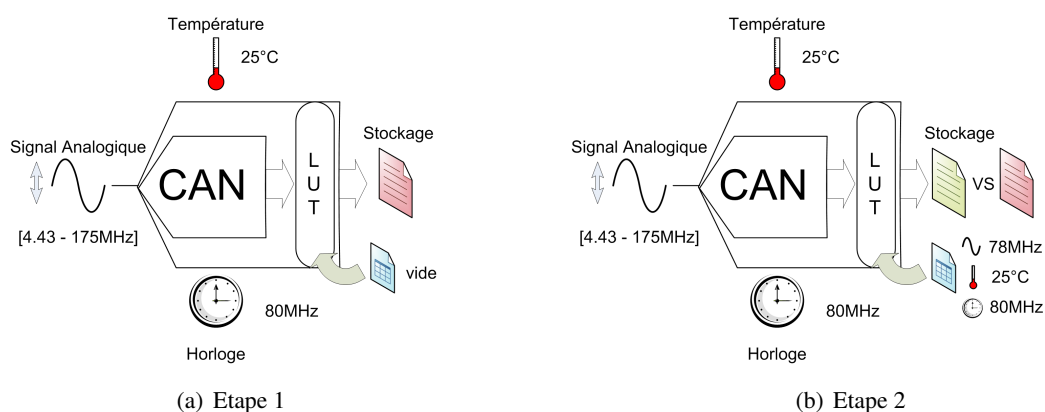


FIG. 4.17 – Compensation avec la LUT calculée à une fréquence d'entrée fixée

Afin de tester le comportement de la compensation en fonction de la fréquence d'utilisation, l'expérience suivante a été menée (figure 4.17) :

- La fréquence d'échantillonnage est fixe pour toute la durée de l'expérience
- La fréquence d'entrée est choisie à une valeur comprise entre 4.43 et 175MHz
- Une acquisition est lancée et les paramètres dynamiques sont stockés pour chaque fréquence appartenant à la plage précitée
- La courbe de non-linéarité est ensuite mesurée pour la fréquence d'entrée fixe de 78MHz
- La table est chargée dans le FPGA et ne sera plus modifiée pour le reste de l'expérience
- Des acquisitions sont lancées pour les différentes fréquences d'entrées précédemment mentionnées, et les nouveaux paramètres dynamiques sont stockés

Les résultats de ces mesures apparaissent sur les figures 4.18. L'amélioration des autres paramètres évolue de façon identique. Pour les fréquences d'entrées inférieures à la fréquence de remplissage (78MHz), la compensation fonctionne correctement, et les améliorations sont légèrement inférieures à celles obtenues précédemment. Par contre, pour les fréquences supérieures à 78MHz, la compensation tend plus rapidement vers les valeurs non corrigées.

La conclusion de cette expérience est que la table calculée pour une fréquence d'entrée de 78MHz permet

4.3. ETUDE DE LA ROBUSTESSE DE LA MÉTHODE DE COMPENSATION

de couvrir un large panel de fréquences, même si l'on observe une limitation de cette correction pour les hautes fréquences d'entrée. Ce phénomène s'explique par le fait que lors du calcul des non-linéarités, les valeurs de gain et d'offset de la courbe sont retirées, or ce sont ces paramètres qui sont le plus dépendants des variations de fréquences d'entrée. Comme l'architecture choisie fait intervenir un échantillonneur-bloqueur, c'est cet étage qui va être le plus sensible aux variations de la fréquence d'entrée. Les problèmes d'appariements sont quant à eux strictement dépendants de la fréquence d'échantillonnage, or celle-ci ne varie pas dans notre expérience. La pratique confirme donc ce que la théorie prévoyait : une table de correction calculée à une fréquence d'entrée précise peut couvrir la bande d'utilisation du convertisseur.

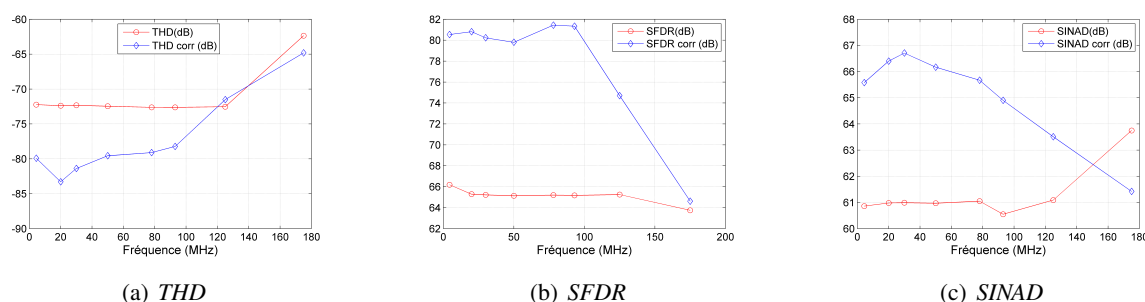


FIG. 4.18 – Évolution de la THD, SFDR et SINAD pour une LUT calculée pour une fréquence d'entrée de 78MHz

4.3.4 Efficacité de la compensation en fonction de la fréquence d'échantillonnage

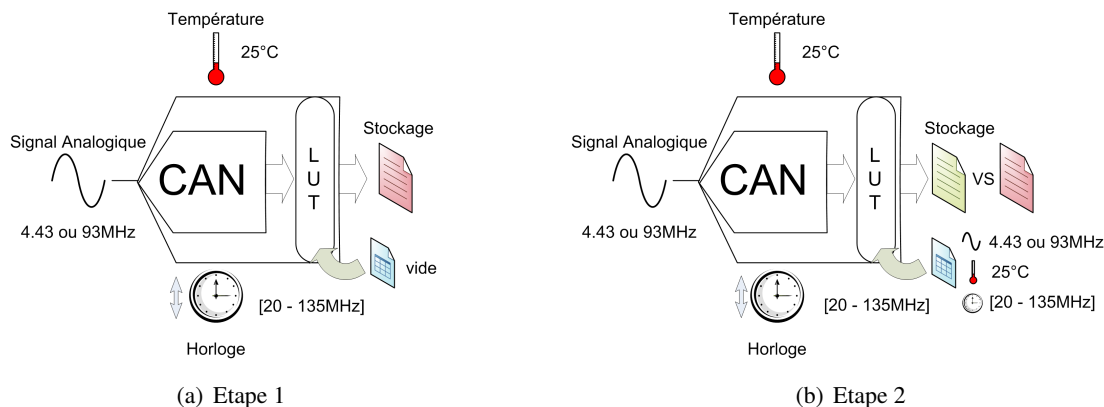


FIG. 4.19 – Compensation avec la LUT calculée en fonction de la fréquence d'échantillonnage

Afin de tester le comportement de la compensation en fonction de la fréquence d'échantillonnage, l'expérience suivante a été menée (figure 4.19) :

- La fréquence d'entrée est choisie à une valeur fixée (4.43MHz ou 93MHz)
- La fréquence d'échantillonnage est choisie dans la plage 20MHz-135MHz
- Une acquisition est lancée et les paramètres dynamiques sont calculés et stockés dans l'interface LabVIEW
- La courbe des non-linéarités est mesurée

- La table est chargée dans le FPGA
- Une nouvelle acquisition est effectuée et les nouveaux paramètres dynamiques sont stockés
- L'opération est réitérée pour une nouvelle fréquence d'échantillonnage

Les résultats de ces mesures apparaissent sur les figures 4.20 et 4.21. La conclusion de cette expérience est que la méthode de compensation fonctionne pour toutes les fréquences d'échantillonnage. Cependant, pour une application spécifique, il est rare que la fréquence d'échantillonnage évolue au cours du temps. La table de correction peut donc être calculée pour une application spécifique, c'est à dire pour une fréquence d'échantillonnage donnée, et une valeur typique de fréquence d'entrée. Le paragraphe précédent a démontré que la table alors calculée pouvait couvrir une large bande de fréquences d'entrée. Bien évidemment, s'il est envisageable de calculer différentes LUT selon l'application, il serait plus flexible d'obtenir une LUT qui puisse couvrir plusieurs fréquences d'entrée et plusieurs fréquences d'échantillonnage.

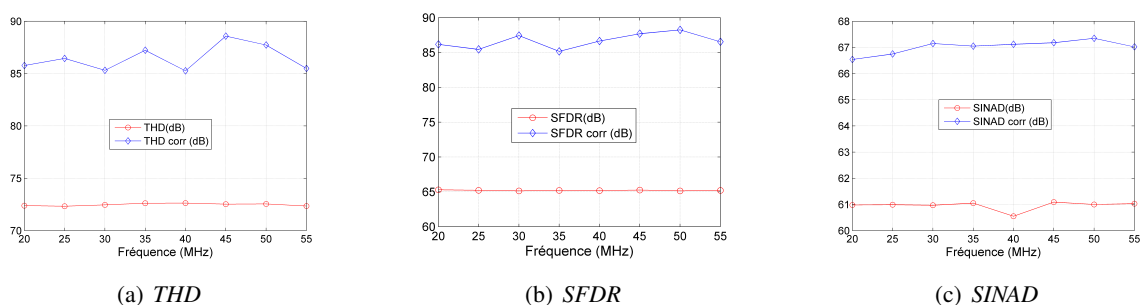


FIG. 4.20 – Évolution de la *THD*, du *SFDR* et du *SINAD* pour une LUT calculée en fonction de la fréquence d'échantillonnage et pour une fréquence d'entrée de 4.43MHz

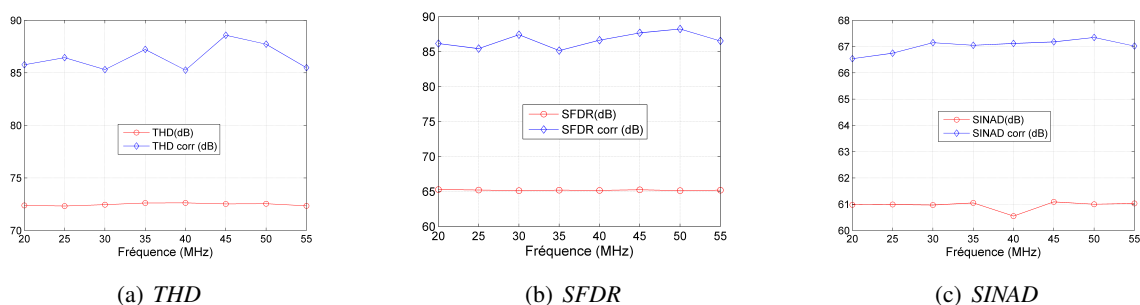


FIG. 4.21 – Évolution de la *THD*, du *SFDR* et du *SINAD* pour une LUT calculée en fonction de la fréquence d'échantillonnage et pour une fréquence d'entrée de 93MHz

Efficacité de la compensation pour une table calculée à une fréquence d'échantillonnage précise

Afin de tester le comportement de la compensation calculée pour une fréquence d'échantillonnage fixe de 60MHz et pour une fréquence d'entrée de 93MHz. L'expérience suivante a été menée (figure 4.22) :

- La fréquence d'entrée est choisie à la valeur fixe de 93MHz
- La fréquence d'échantillonnage est choisie dans l'intervalle 10MHz-120MHz

4.3. ETUDE DE LA ROBUSTESSE DE LA MÉTHODE DE COMPENSATION

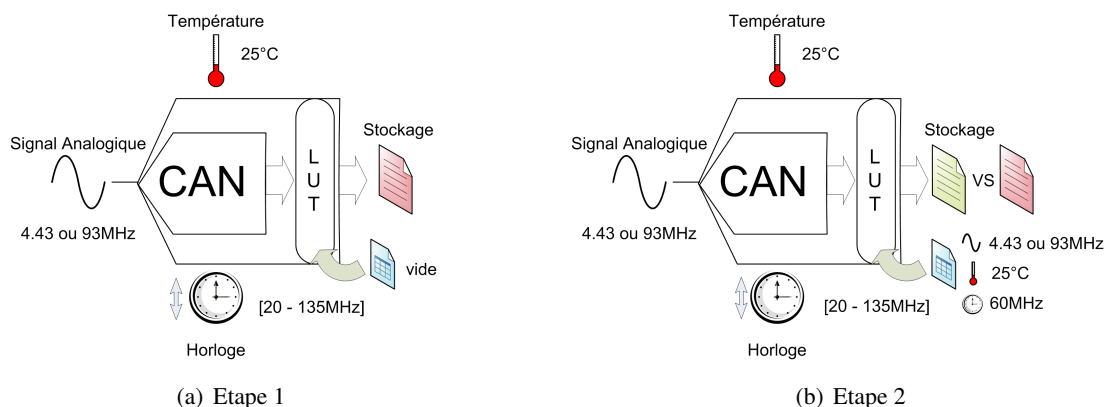


FIG. 4.22 – Compensation avec la LUT calculée pour des fréquences d'entrée et d'échantillonnage fixées

- Une acquisition est lancée et les paramètres dynamiques sont calculés et stockés dans l'interface LabVIEW
- La fréquence d'échantillonnage est modifiée jusqu'à ce que tout l'intervalle soit couvert et les paramètres stockés
- La fréquence d'échantillonnage est ensuite choisie à la valeur fixe de 60MHz
- La courbe des non-linéarités est mesurée
- La table est chargée dans le FPGA
- La fréquence d'échantillonnage est modifiée, toujours dans l'intervalle considéré
- Une nouvelle acquisition est effectuée et les nouveaux paramètres dynamiques sont stockés
- L'opération est répétée jusqu'à ce que tout l'intervalle soit couvert et les nouveaux paramètres stockés

Les résultats de ces mesures apparaissent sur les figures 4.23. Les paramètres *SFDR*, *THD* et *SINAD* possèdent une valeur optimale pour la fréquence d'échantillonnage correspondant à celle du calcul de la table, mais conservent tout de même une amélioration significative lors de l'évolution fréquentielle.

La conclusion de cette expérience est que la méthode de compensation fonctionne pour un calcul de table effectué à une fréquence d'échantillonnage fixe. Cependant, la qualité de la compensation diminue lorsque l'on s'éloigne de la fréquence de référence. L'obtention d'une même table de correction susceptible de couvrir un large panel d'applications semble donc difficile.

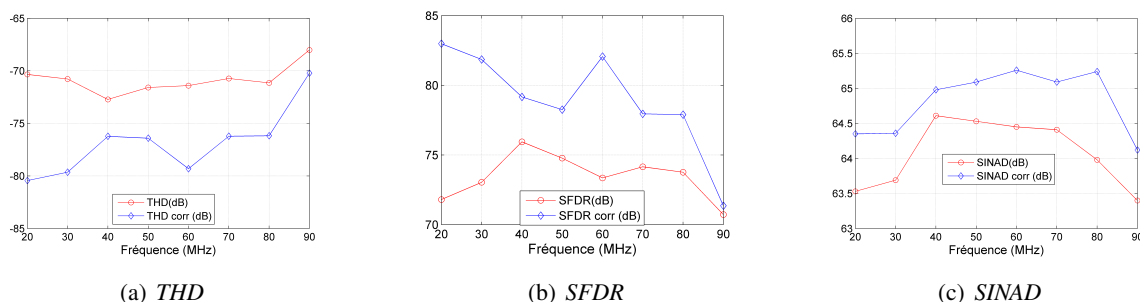


FIG. 4.23 – Évolution de la *THD*, du *SFDR* et du *SINAD* en fonction de la fréquence d'échantillonnage pour une LUT calculée à une fréquence d'échantillonnage de 60MHz et pour une fréquence d'entrée de 93MHz

Efficacité de la compensation en fonction de la Température

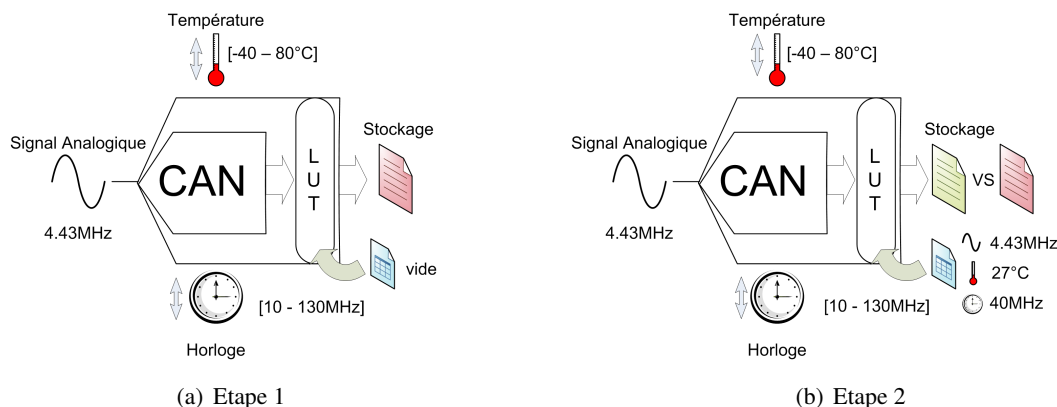


FIG. 4.24 – Compensation en fonction de la température et de la fréquence d'échantillonnage pour une LUT fixée

Un autre paramètre à prendre en compte dans l'utilisation du convertisseur, est son comportement vis-à-vis de la température. L'expérience suivante a été menée :

- La fréquence d'échantillonnage est choisie dans l'intervalle 10MHz-130MHz
- La fréquence d'entrée est fixée à une valeur de 4.43MHz
- Le circuit est mis à la température $T \in [-40..80]^{\circ}C$ à l'aide d'un thermostat
- Une acquisition est lancée et les paramètres dynamiques sont calculés et stockés dans l'interface LabVIEW
- La fréquence d'échantillonnage et la température sont modifiées jusqu'à ce que tous les intervalles soient couverts et les paramètres stockés
- La fréquence d'échantillonnage est ensuite choisie à la valeur fixe de 40MHz
- La température est fixée à $27^{\circ}C$
- La courbe des non-linéarités est mesurée
- La table est chargée dans le FPGA et ne sera plus modifiée pour le reste de l'expérience
- La fréquence d'échantillonnage est modifiée, toujours dans l'intervalle considéré
- La température est modifiée au sein de son intervalle d'évolution
- Une nouvelle acquisition est effectuée et les nouveaux paramètres dynamiques sont stockés
- L'opération est répétée jusqu'à ce que tous les intervalles soient couverts et les nouveaux paramètres stockés

Les résultats de ces mesures apparaissent sur les figures 4.25 à 4.27. La température est placée en abscisse, tandis que la fréquence d'échantillonnage est placée en ordonnée. Cette visualisation permet de comparer aisément les figures avant compensation (situées à gauche) de celles après compensation (situées à droite). Plus la couleur est proche du bleu, moins bonnes sont les performances. A l'inverse, plus la couleur est proche du rouge, meilleures sont les performances.

On peut observer que plus la fréquence d'échantillonnage et la température augmentent, plus les performances se dégradent. En revanche, lorsque la fréquence d'échantillonnage et la température avoisinent les valeurs de références utilisées pour le calcul de la table de compensation, on observe une zone de bonnes performances. Globalement, même si les améliorations ne sont pas tout le temps optimales, elles

4.3. ETUDE DE LA ROBUSTESSE DE LA MÉTHODE DE COMPENSATION

restent tout de même positives.

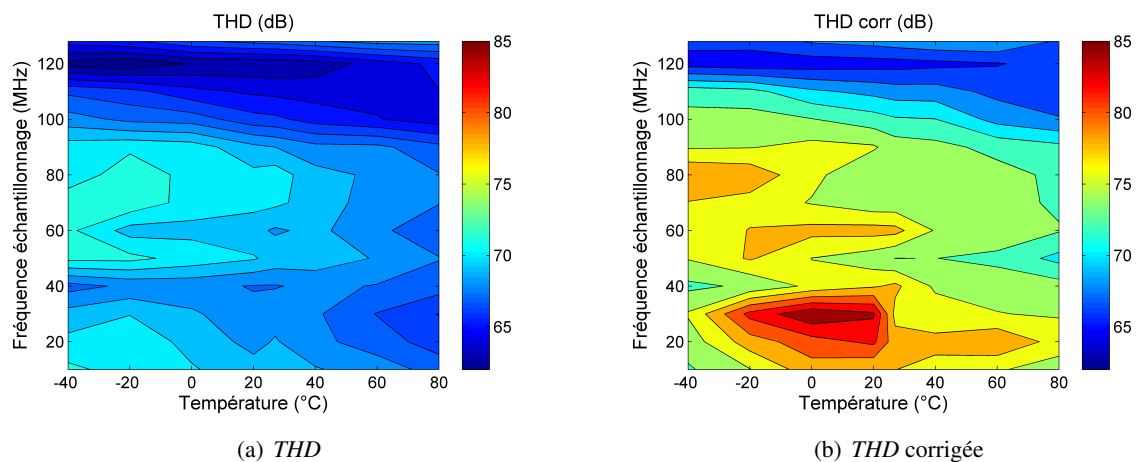


FIG. 4.25 – Évolution des valeurs de *THD* en fonction de la température et de la fréquence d'échantillonnage pour une LUT remplie à 40MHz et à une température de 27°C

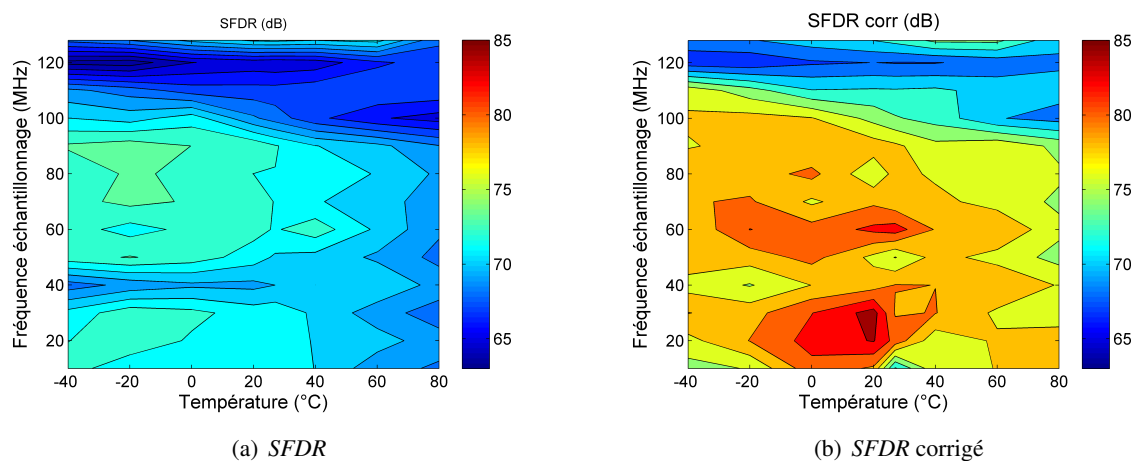


FIG. 4.26 – Évolution des valeurs de *SFDR* en fonction de la température et de la fréquence d'échantillonnage pour une LUT remplie à 40MHz et à une température de 27°C

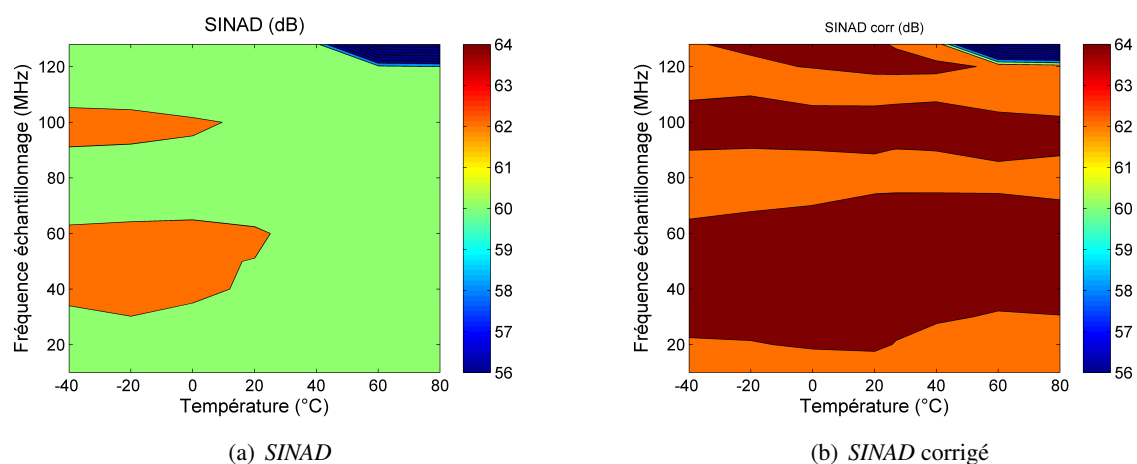


FIG. 4.27 – Évolution des valeurs de *SINAD* en fonction de la température et de la fréquence d'échantillonnage pour une LUT remplie à 40MHz et à une température de 27°C

4.4 Conclusion

Dans ce chapitre, le véhicule de test a été expliqué ainsi que les différentes procédures utilisées pour tester ce véhicule. En se plaçant dans le cadre d'une compensation *offline*, nous avons pu conclure que le choix de la méthode importait peu car elles fournissaient les mêmes ordres de grandeur de correction. Une étude sur la robustesse de la méthode de la compensation vis-à-vis des variations en fréquences et en températures a également été menée. Son aboutissement permet de conclure que pour des valeurs des fréquences d'entrée et d'échantillonnage et de température fixées, une table de correspondance peut être calculée et compenser les non-linéarités du convertisseur sur une large gamme d'applications.

4.4. CONCLUSION

Chapitre 5

Autocompensation embarquée

Dans ce chapitre, la méthode de compensation est adaptée pour être intégrée directement au sein du composant. Après l'énoncé des besoins et des contraintes, nous proposons d'étudier différentes solutions de réduction de la complexité et de temps de calcul de la table en fonction de l'architecture utilisée. Enfin, nous nous attardons sur les méthodes embarquées de calcul des harmoniques du signal converti.

5.1 Présentation de la compensation embarquée

Dans le chapitre précédent nous avons pu juger de l'efficacité de la compensation de type *offline*. Même si cette méthode est robuste vis à vis des fréquences d'utilisation et de la température, elle impose une table de correction figée pour la vie du produit. Cette table ne peut être recalculée, ni affinée, selon l'emploi du produit et donc la compensation peut ne plus être optimale sur toutes les plages d'utilisation. Pour obtenir l'optimum de correction quelques soient les conditions d'utilisation ou les dérives dues au vieillissement du composant, il faudrait pouvoir calculer la table à chaque nouvelle utilisation, par exemple au démarrage de la puce : on parle alors de compensation embarquée, ou compensation *online*. La compensation *online* a pour but de s'adapter aux divers modes de fonctionnement du convertisseur, quelques soient les fréquences utilisées, quelque soit l'âge du produit, et quelque soit la température de fonctionnement. Afin de pouvoir calculer les nouvelles tables à la demande, il faut pouvoir embarquer au sein du convertisseur l'ensemble des composants nécessaires à la mesure des non-linéarités et au remplissage de la table de correction. La figure 5.1 représente la configuration idéale à embarquer pour être capable de mesurer les valeurs correctives.

Il y a trois organes essentiels à l'obtention d'une compensation embarquée :

- **Le générateur de stimuli** : afin de pouvoir mesurer les erreurs issues du convertisseur à corriger, il faut fournir à l'entrée de ce dernier un signal aux propriétés connues. Jusqu'à présent, nous utilisons pour nos mesures un signal issu d'un synthétiseur de signaux sinusoïdaux, couplé avec un filtre passe bande, et dont la fréquence était calculée de façon cohérente avec la fréquence d'échantillonnage. De plus, afin d'assurer une synchronisation de ces deux signaux, un signal basse fréquence reliait les deux appareils. Pour une compensation embarquée, il devient donc nécessaire d'inclure au sein du composant un générateur de stimuli pour pouvoir fonctionner de façon autonome. Une façon usuelle

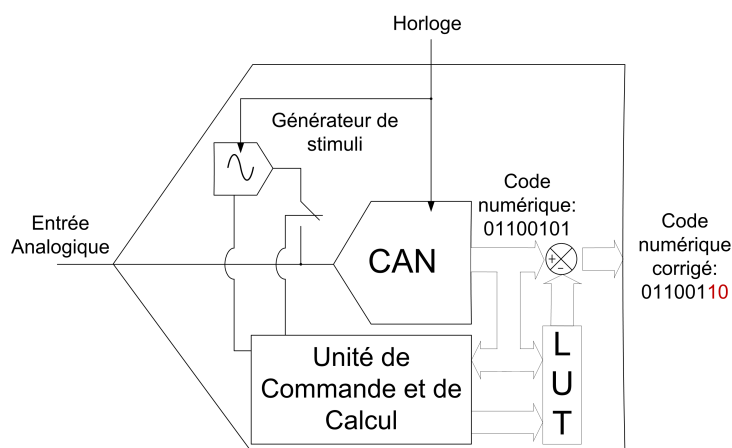


FIG. 5.1 – Configuration idéale de correction

de générer un signal sinusoïdal est d'utiliser un convertisseur $\Sigma - \Delta$ [RDM04], auquel on peut associer un filtre de type SAW [HJ85] pour obtenir un signal proche du cas idéal. Ce générateur ne sera pas traité dans le cas de cette étude. Nous nous contenterons d'utiliser l'entrée analogique classique du convertisseur afin de nous consacrer aux deux autres organes. D'autres travaux sont nécessaires afin d'étudier la faisabilité de cette génération, travaux qu'il faut coupler aux besoins de la méthode de mesure. En effet, dans le cadre d'une mesure par histogramme, le stimulus n'est pas obligatoirement sinusoïdal, une rampe peut également être utilisée. Les travaux menés dans [ABB⁺01, BABR02] vont dans ce sens.

- **L'unité de commande et de calcul** : cet organe, entièrement numérique, se charge de gérer les différentes phases nécessaires à la mesure, au calcul des non-linéarités et au remplissage de la table de correspondance. Il doit pouvoir contrôler la génération de stimuli, et doit ensuite procéder au calcul des erreurs. Une fois cette phase accomplie, la LUT est mise à jour avec les nouvelles valeurs. Comme indiqué précédemment, le générateur de stimuli ne sera pas étudié ici, l'unité de commande et calcul devra donc se limiter au calcul et au remplissage de la table.
- **La table de correspondance LUT** : la table stocke les valeurs correctives à appliquer aux codes de sorties du convertisseur. Elle est remplie par l'unité de commande et de calcul. Son fonctionnement est en tout point similaire à celui exposé dans le chapitre précédent.

Parmi tout ces organes importants, ce chapitre sera consacré à l'unité de commande et de calcul. Nous nous attarderons essentiellement sur les contraintes liées au calcul des non-linéarités.

5.1.1 Unité de commande et de calcul

L'unité de commande et de calcul est l'organe à optimiser au sein du système embarqué. En effet, la taille de la table de correspondance a une taille figée : il faut donc être en mesure de proposer une architecture de mesure et de calcul qui utilise le moins de ressources possibles. Les ressources en question sont principalement liées à la surface de silicium utilisée. Plus cette taille sera petite, moins cher sera le produit. Nous tenterons d'établir une comparaison en terme de nombre de portes logiques utilisées entre les différentes méthodes proposées, même s'il est très difficile d'établir un budget vis à vis de la

consommation et de la surface ajoutée.

La figure 5.2 résume les fonctions de base que l'unité de calcul doit pouvoir effectuer. A partir des données issues de la conversion, une extraction des non-linéarités est réalisée afin de pouvoir fournir les valeurs correctives à la table de correspondance.

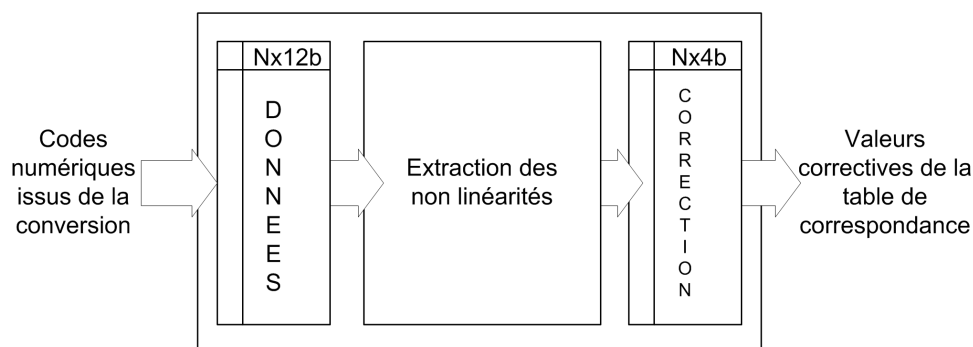


FIG. 5.2 – Fonctions de l'unité de calcul

Le chapitre 3 a mis en avant les différentes techniques d'extraction des non-linéarités. Il a été conclu dans ce chapitre que la méthode par histogramme réclamait un nombre de points trop important (et donc une taille mémoire trop importante) au regard du nombre de points requis par les méthodes spectrales. Parmi les méthodes spectrales, celles utilisant des polynômes s'avèrent être trop imprécises pour être utilisées pour la compensation. Notre choix se tourne donc vers la nouvelle méthode utilisant une décomposition en série de Fourier de la courbe de non linéarités.

Deux axes majeurs ressortent de l'utilisation de cette dernière technique en tant que technique d'extraction embarquée :

- La taille de la matrice $T_{K_{MAX}}$ utilisée doit pouvoir être optimisée vis à vis de la connaissance de l'architecture. En effet, l'utilisation d'une architecture à repliement et interpolation engendre une forme de courbe des non-linéarités spécifique. Une étude approfondie de cette relation permettrait de sélectionner les paramètres décrivant la courbe de non-linéarités et donc de simplifier la forme de la matrice. Cette étude constitue le propos de la section suivante.
- L'extraction des non-linéarités du convertisseur, s'effectue à travers l'analyse spectrale du signal converti. De façon classique, l'analyse spectrale se réalise à partir de l'opération Transformée de Fourier, or il s'avère que cette opération est très gourmande en ressource silicium. D'autres méthodes moins gourmandes doivent donc être étudiées. C'est le propos de la dernière section qui compare l'analyse par transformée de Fourier, le calcul par l'algorithme de Goertzel ainsi qu'une nouvelle méthode développée dans le cadre de cette étude.

Les sections suivantes présentent de nombreuses avancées dans la compréhension du processus global d'autocompensation. Cependant, l'objectif final n'a pas été atteint, et ces avancées doivent être considérées comme la base de futures études.

Le lecteur avide de réaliser une compensation embarquée trouvera ici de nombreuses pistes permettant de faire évoluer son projet.

5.2 Réduction de la complexité de la méthode spectrale

La méthode spectrale choisie pour l'extraction des non-linéarités est celle décrite dans la section 3.3.2 de ce manuscrit. Cette méthode permet de relier par une équation matricielle, la description fréquentielle du signal converti $s(\cdot)$ et la description fréquentielle de la courbe des non-linéarités périodisée $\Phi(\cdot)$. Nous allons dans un premier temps nous attarder sur la modélisation fréquentielle de la fonction $\Phi(\cdot)$ dans le cadre d'une architecture à repliement et interpolation. Ensuite, nous mettrons en relation la nouvelle représentation fréquentielle et son impact sur la forme de la matrice $T_{K_{MAX}}$. Finalement, nous poursuivrons sur l'impact de la matrice sur le choix des harmoniques à prendre en considération.

5.2.1 Modélisation de la courbe des non-linéarités pour une architecture à repliement et interpolation

L'architecture à repliement et interpolation a été décrite en détail au chapitre 1. La nature des diverses erreurs des composants de cette architecture a également été décrite de manière ponctuelle, c'est à dire que les dérives du composant ont été expliquées, mais que l'impact global sur la forme de la courbe des non-linéarités n'a pas été exposé. Nous ne tenterons pas ici de relier cet impact directement aux différentes sources d'erreurs exposées au chapitre 1 : cette étude réclamerait de suivre avec rigueur la conception physique de chaque bloc de l'architecture, multipliant alors le nombre de paramètres à prendre en considération. Dans cette section, nous nous contenterons d'observer la signature de l'architecture sur la courbe de non-linéarités. Le lecteur désireux de modéliser plus en détails les erreurs de l'architecture est convié à se référer aux travaux menés par [LSMW⁺04]. De nombreuses pistes sont fournies au regard du produit développé par l'auteur.

La figure 5.3 illustre le principe de la décomposition de la fonction de transfert énoncée au chapitre 1. Cette décomposition se traduit par une première conversion grossière à partir d'un convertisseur de résolution Q . A partir de cette première conversion, un signal résidu est obtenu et est dirigé vers un convertisseur de résolution q .

Si l'on considère que le premier convertisseur est idéal, alors sa courbe de non-linéarités vaut 0 quelque soit le code excité (figure 5.4(a)). Si l'on considère que le second convertisseur n'est pas idéal, alors il possède une courbe de non-linéarités non nulle dépendante du code excité (figure 5.4(b)). La composition de ces deux fonctions de transfert fournit le code de sortie du convertisseur. Le fait qu'uniquement le second convertisseur possède des non-linéarités se traduit par la courbe de non linéarités globales illustrée sur la figure 5.5. La récurrence d'apparition des non-linéarités du convertisseur de résolution q est évidente.

En observant une courbe de non linéarités issue d'un convertisseur réel (figure 5.6), l'apparition de ces récurrences est également évidente. Sur la figure, une sinusoïde de période 128 codes a été tracée afin de mettre en évidence ce phénomène cyclique. Le nombre 128 provient du fait que l'architecture utilise 64 signaux de référence dans le premier étage de repliement. Comme la résolution du convertisseur est de 12bits, le convertisseur de résolution q convertit $2^{12}/64 = 64$ niveaux différents entre chaque tension de référence.

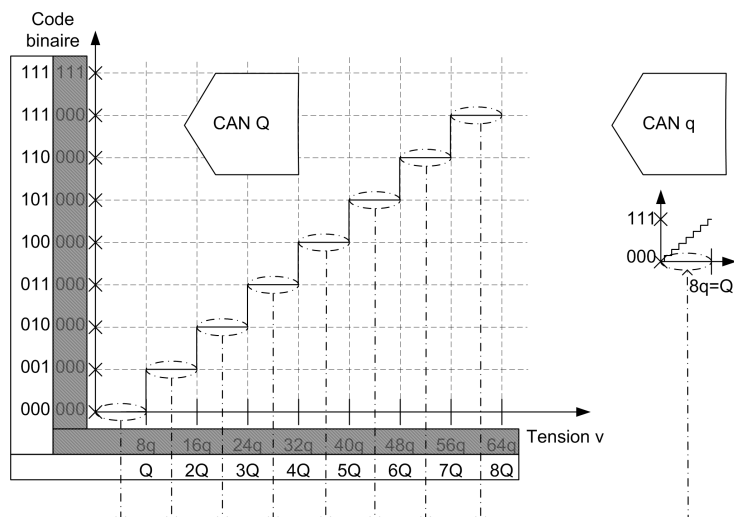


FIG. 5.3 – Fonctions de transfert

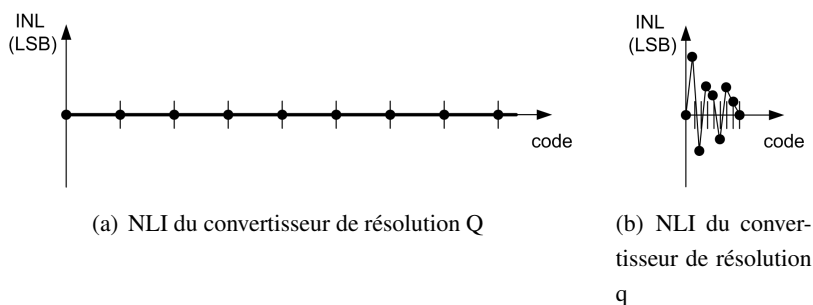


FIG. 5.4 – Courbes de NLI des deux convertisseurs de l'architecture

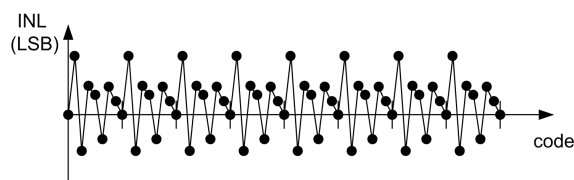


FIG. 5.5 – NLI résultante

Pour faire apparaître la présence d'autres cycles au sein de cette courbe de non linéarités, l'utilisation d'une transformée de Fourier est toute indiquée. Afin de ne pas confondre l'utilisation de la transformée de Fourier pour le signal converti et pour la courbe des non-linéarités, nous utiliserons un nouveau formalisme. Ce formalisme décrit dans la table 5.2.1 transpose les notions bien établies de temps, période, fréquence et spectre en un nouveau champ lexical.

Lorsqu'une transformée de Fourier est appliquée sur une courbe de non-linéarités (figure 5.7(a)), les récurrences apparaissent clairement (figure 5.7(b)), et les cycles peuvent alors être déterminés. Par exemple, la récurrence $R = 32$ est prédominante dans le spectr_inl de la figure 5.7(b), elle correspond à un cycle de $C = 2^{12}/32 = 128$. Il s'agit bien du cycle observé sur la figure 5.6.

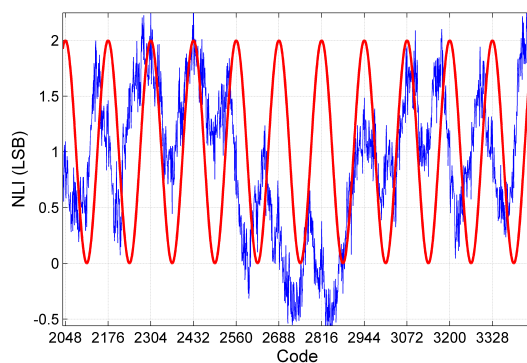
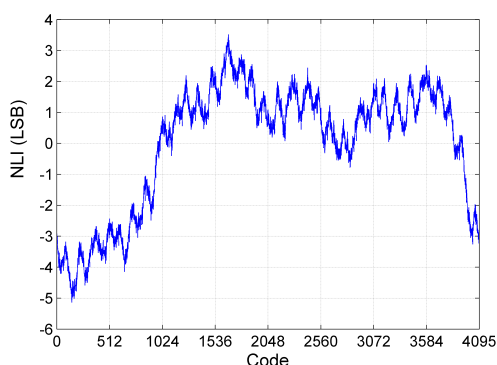


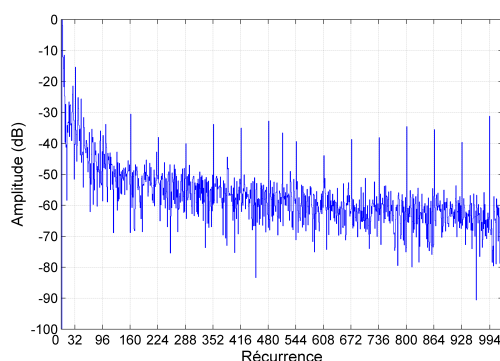
FIG. 5.6 – Cyclicité de la NLI

Signaux usuels	Courbe de non linéarités
Temps (t)	Code (c)
Période (T)	Cycle (C)
Fréquence (f)	Récurrence (R)
Spectre	Spectr_inl

TAB. 5.1 – Formalisme de l'analyse fréquentielle des NLI



(a) NLI



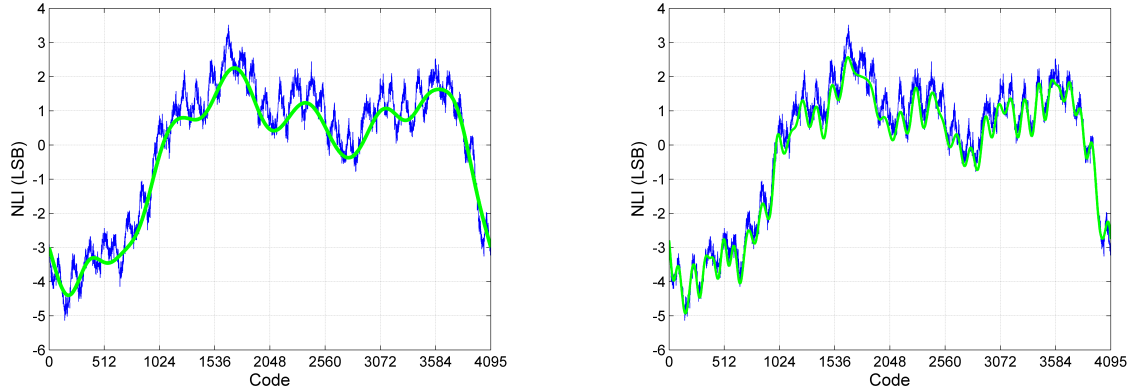
(b) Récurrences de le NLI

FIG. 5.7 – Récurrences au sein de la courbe de non-linéarités

De la même façon que pour un spectre classique, le spectr_inl traduit la puissance et donc la présence de certaines récurrences au sein de la courbe transformée. Plus la puissance de la récurrence est élevée, plus le cycle sera visible au sein de la forme de la courbe. En utilisant uniquement les récurrences de puissances fortes, on peut ainsi obtenir une description optimisée de la courbe de non-linéarités. La figure 5.7(b) met en évidence la prépondérance de la récurrence $R = 32$ et de ses harmoniques impaires $R = (2 \times k + 1) \times 32$. Les premières récurrences (< 32) possèdent également une énergie importante, elles sont donc à considérer dans le modèle de description.

La figure 5.8 met en évidence la différence de reconstruction de la courbe de non-linéarités en fonction des récurrences choisies pour décrire la fonction. Ainsi la figure 5.8(a) représente la courbe des non-linéarités reconstruite à partir des seules 28 premières récurrences, tandis que la figure 5.8(b) représente

la courbe reconstruite lorsque les 28 récurrences les plus puissantes ont été sélectionnées.



(a) NLI reconstruite à partir des 28 premières récurrences

(b) NLI reconstruite à partir de 28 récurrences choisies

FIG. 5.8 – Reconstruction en fonction du choix des récurrences

En appliquant ces nouvelles considérations à la formule décrivant la fonction périodique $\Phi(\cdot)$ énoncée au chapitre 3, on obtient une description simplifiée de la fonction de non-linéarités en fonction du nombre K_p des premières récurrences, et du nombre K_h d'harmoniques considérées :

$$\Phi(x) \approx \frac{a_0}{2} + \sum_{k=1}^{K_p} (a_k \cos(k\omega x) + b_k \sin(k\omega x)) + \sum_{k=0}^{K_h} (a_{(2 \times k + 1) \times 32} \cos(((2 \times k + 1) \times 32)\omega x) + b_{(2 \times k + 1) \times 32} \sin(((2 \times k + 1) \times 32)\omega x)) \quad (5.1)$$

Cette nouvelle modélisation optimisée de la fonction représentant les non-linéarités du convertisseur réduit le nombre de paramètres nécessaires à une description correcte de la fonction. L'impact de ce jeu de paramètres réduits sur le calcul de la matrice $T_{K_{MAX}}$ est exprimé dans la section suivante.

5.2.2 Impact de l'architecture sur la matrice $T_{K_{MAX}}$

Dans le chapitre 3, la relation suivante, reliant les harmoniques paires issues du signal converti aux paramètres a_k décrivant la fonction de non-linéarités $\Phi(\cdot)$, a été établie :

$$\begin{bmatrix} S_0^{Th} \\ S_2^{Th} \\ \vdots \\ S_{H_{max}-1}^{Th} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & A_1^0 & \dots & A_{K_{max}}^0 \\ 0 & A_1^2 & \dots & A_{K_{max}}^2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & A_1^{H_{max}-1} & \dots & A_{K_{max}}^{H_{max}-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K_{max}} \end{bmatrix} \quad (5.2)$$

5.2. RÉDUCTION DE LA COMPLEXITÉ DE LA MÉTHODE SPECTRALE

A partir de la nouvelle description, cette relation peut être optimisée de la façon suivante, afin de ne tenir compte que des paramètres majeurs :

$$\begin{bmatrix} S_0^{Th} \\ S_2^{Th} \\ \vdots \\ S_{H_{max}-1}^{Th} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & A_1^0 & \dots & A_{K_p}^0 & A_{32}^0 & \dots & A_{(2k+1) \times 32}^0 & \dots & A_{(2K_h+1) \times 32}^0 \\ 0 & A_1^2 & \vdots & A_{K_p}^2 & A_{32}^2 & \dots & A_{(2k+1) \times 32}^2 & \dots & A_{(2K_h+1) \times 32}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & A_1^{H_{MAX}} & \dots & A_{K_p}^{H_{MAX}} & A_{32}^{H_{MAX}} & \dots & A_{(2k+1) \times 32}^{H_{MAX}} & \dots & A_{(2K_h+1) \times 32}^{H_{MAX}} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K_p} \\ a_{32} \\ \vdots \\ a_{(2k+1) \times 32} \\ \vdots \\ a_{(2K_h+1) \times 32} \end{bmatrix} \quad (5.3)$$

Cette forme, bien que plus compliquée à écrire, est optimisée vis-à-vis du type de convertisseur à compenser. Une première étude des non-linéarités du convertisseur permet d'analyser les récurrences identifiant la courbe. A partir de cette analyse, nous sommes à même de réduire le nombre K_{MAX} de paramètres complexes nécessaires, en un ensemble réduit de taille $K_p + K_h$. La figure 5.9 illustre cette réduction. En omettant les paramètres de faible influence, la matrice $T_{K_{MAX}}$ est alors grandement réduite. D'une taille 600×600 , elle se trouve réduite à la taille 40×40 . Les calculs sont donc plus simples à réaliser.

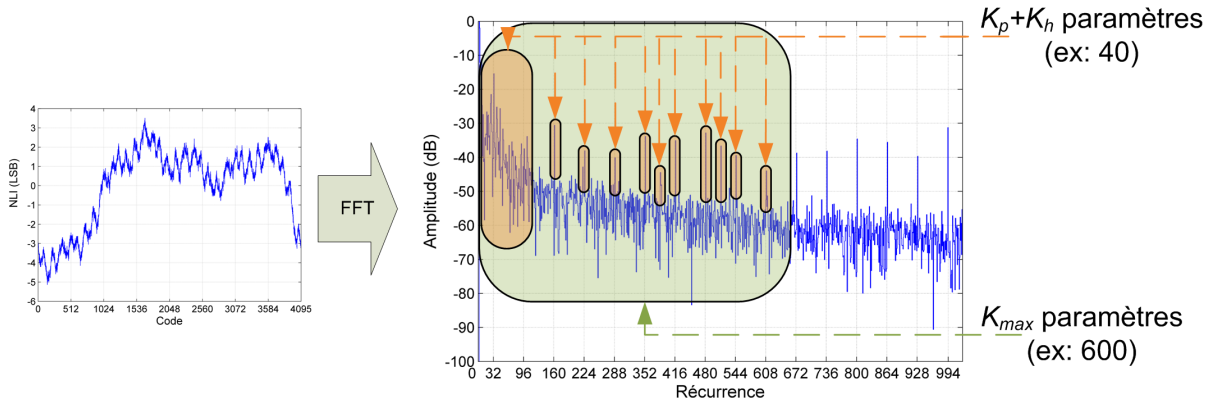


FIG. 5.9 – Réduction du nombre de paramètres

5.2.3 Impact de l'architecture sur le choix des Harmoniques du signal converti

L'influence d'une courbe de non-linéarité précise se traduit par l'apparition de diverses harmoniques sur le spectre du signal converti. Dans notre exemple nous avons simulé l'influence d'une courbe de NLI représentée par un cosinus de période 5ω (paramètre a_5 dans l'équation de $\Phi(\cdot)$) sur le spectre de sortie. La figure 5.10(a) représente la courbe de non-linéarités, tandis que la figure 5.10(b) illustre les effets de cette NLI précise. Les numéros des harmoniques apparaissent sur cette dernière figure, et comme notre

méthode spectrale permettait de le prévoir, seules les harmoniques paires apparaissent sur ce spectre replié. Les 18 premières harmoniques paires sont présentes, cependant, ce n'est pas la raie numéro 2 qui est la plus élevée. La raie la plus élevée est celle de l'harmonique 14 suivie par les harmoniques 4, 6, 10 et 16.

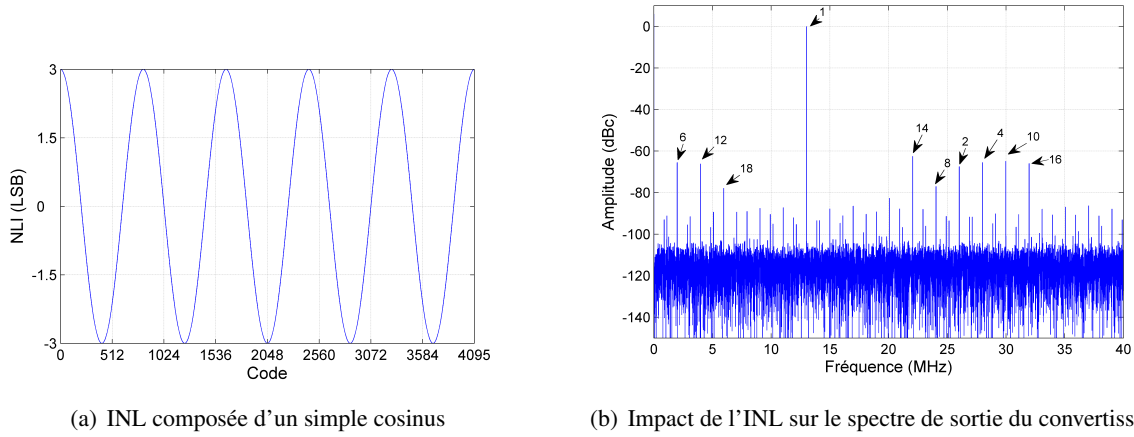


FIG. 5.10 – Influence d'un cosinus

Afin de comprendre la différence d'énergie des raies harmoniques, il faut se pencher sur la représentation matricielle de notre modèle. Dans le chapitre 3, les vecteurs colonnes A_k et B_k ont été décrits. Ils composent les colonnes de la matrice $T_{K_{MAX}}$. Lorsque la courbe des non-linéarités est uniquement décrite par le paramètre a_k , alors seul le vecteur colonne A_k intervient dans le produit matriciel. Ci dessous, nous simplifions l'écriture du produit lorsque seul le paramètre a_5 intervient ce qui est le cas dans l'exemple considéré.

$$\begin{bmatrix} S_0^{Th} \\ S_2^{Th} \\ \vdots \\ S_{H_{max}-1}^{Th} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & A_1^0 & \dots & A_{K_{max}}^0 \\ 0 & A_1^2 & \dots & A_{K_{max}}^2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & A_1^{H_{max}-1} & \dots & A_{K_{max}}^{H_{max}-1} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_5 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.4)$$

Ce produit peut finalement se simplifier par la multiplication d'un simple scalaire et d'un vecteur colonne.

$$\begin{bmatrix} S_0^{Th} \\ S_2^{Th} \\ \vdots \\ S_{H_{max}-1}^{Th} \end{bmatrix} = \begin{bmatrix} A_5^0 \\ A_5^2 \\ \vdots \\ A_5^{H_{max}-1} \end{bmatrix} \cdot a_5 \quad (5.5)$$

En observant graphiquement l'allure de ce vecteur colonne, on peut remarquer des variations correspondant bien aux différences d'énergie des raies harmoniques (figure 5.11). On retrouve bien l'ordre des

5.2. RÉDUCTION DE LA COMPLEXITÉ DE LA MÉTHODE SPECTRALE

raies harmoniques de la figure 5.10(b), à savoir : la raie de plus forte énergie est l'harmonique 14, suivie des harmoniques 4,6,10 et 16.

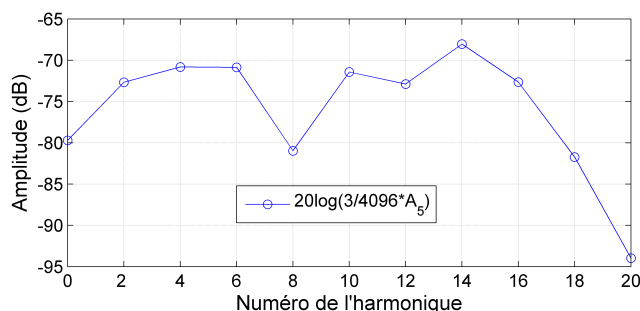


FIG. 5.11 – Allure du vecteur A_5

Cette information est très importante dans le cadre de notre estimation. En effet, au regard des différences de valeurs d'énergies des raies, l'erreur de mesure que l'on peut effectuer sur l'harmonique 2 (-72dB) est plus importante que l'erreur de mesure que l'on peut effectuer sur l'harmonique 14 (-68dB). Par exemple, dans le cas où $a_5 = 3$, lorsque l'on effectue le calcul inverse, une erreur de 2dB sur la valeur de H_2 nous fournit la valeur 4.7, alors qu'une erreur de 2dB sur H_{14} nous fournit la valeur 3.7. Mieux vaut donc connaître à l'avance l'harmonique susceptible de contenir le plus d'information.

La figure 5.12 représente l'allure du vecteur A_{199} pour une valeur de a_{199} de 1LSB. L'influence de ce paramètre se traduit par l'apparition d'une raie harmonique d'ordre supérieure à 600. La contribution de cette raie est de -87dB, alors que les contributions des premières raies harmoniques ont un niveau inférieur à -95dB. Ce niveau très faible ne nous permet pas d'obtenir une mesure fiable de l'harmonique considérée, et donc, l'estimation de la valeur du paramètre a_{199} s'en trouvera grandement faussée. Par contre, si l'on mesure la raie de plus forte contribution, l'erreur effectuée sur sa mesure nous permettra d'obtenir une estimation plus fiable du paramètre a_{199} . Dans la réalité, l'observation d'un phénomène cyclique de l'ordre de 199 codes ayant une amplitude de 1LSB n'a que très peu de chances d'apparaître. L'apparition de la raie harmonique d'ordre supérieure à 600 a donc elle aussi, très peu de chance de se réaliser.

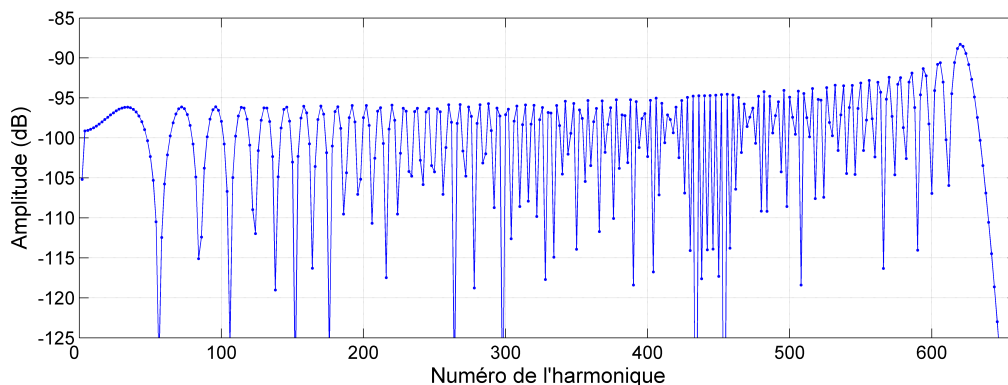


FIG. 5.12 – Allure du vecteur A_{199}

Lors de la sélection des paramètres a_k et b_k décrivant la courbe de non-linéarité, il faut observer la position de la valeur maximale du vecteur A_k ou B_k afin d'être sûr que la raie que l'on va tenter de mesurer dans le spectre du signal converti contienne bien le maximum d'informations vis-à-vis des paramètres choisis. La figure 5.13 illustre la méthode de sélection des harmoniques à mesurer. Après avoir obtenu une représentation moyennée de la courbe des non-linéarités d'un convertisseur (1), on applique une opération de transformée de Fourier afin d'obtenir le Spectr_inl (2). A partir de ce Spectr_inl, nous sommes à même de choisir les valeurs des paramètres a_k et b_k . A partir de ces valeurs, il nous faut étudier l'allure des vecteurs A_k et B_k . Dans notre exemple, nous étudions le vecteur A_5 (3) associé au paramètre a_5 . Sur cette figure, nous observons que l'harmonique qui est la plus influencée par ce paramètre est l'harmonique 14. Nous choisissons donc de mesurer l'harmonique 14 du signal converti (4). Ainsi, nous réduisons l'erreur d'estimation du paramètre a_5 lorsque nous procédons à l'opération inverse, à savoir, estimer le paramètre a_5 à partir des valeurs mesurées sur le spectre du signal converti.

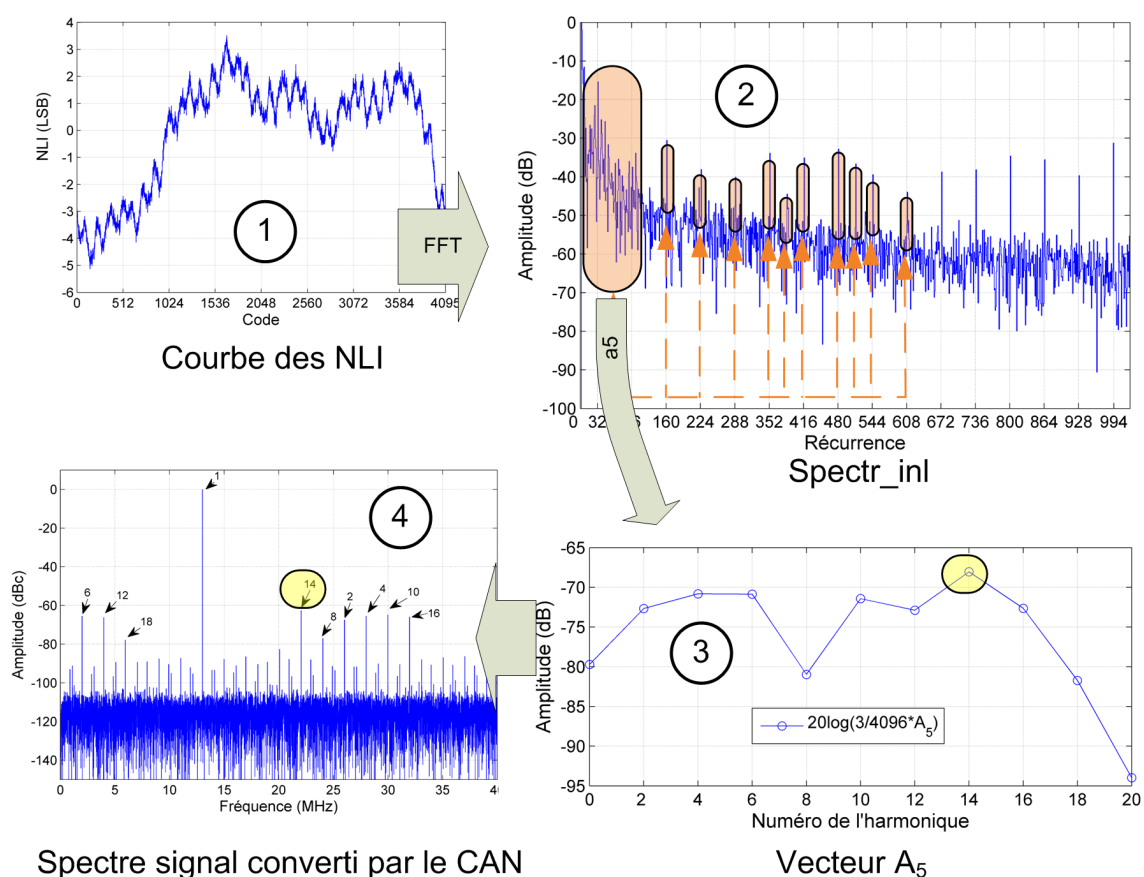


FIG. 5.13 – Sélection des harmoniques à mesurer

5.2.4 Conclusion

Dans cette section, nous avons pu réduire la complexité de la méthode spectrale par une connaissance approfondie de l'impact de l'architecture du convertisseur. Dans les architectures redondantes, telles que

celles à repliement et interpolation, des cycles de non-linéarités apparaissent et permettent de simplifier le modèle d'extraction. Afin de s'assurer d'obtenir une estimation optimale, un soin particulier doit être porté au choix des mesures des harmoniques du signal converti. La partie suivante traite de la façon de mesurer les harmoniques en question.

5.3 Calcul des harmoniques In Situ

Dans le chapitre 3, l'extraction des non-linéarités du convertisseur s'effectue au travers d'une analyse spectrale. Le nombre d'harmoniques nécessaires à une estimation de ces non-linéarités à été réduit dans les sections précédentes grâce à une compréhension plus détaillée des architectures à redondance, telles que les architectures à repliement et à interpolation. Dans cette section, nous nous focaliserons sur les différentes méthodes permettant d'obtenir les valeurs des puissances des raies harmoniques choisies. Le contexte de l'étude étant la compensation *online*, nous nous attacherons à consommer le moins possible de ressources de calcul afin de réduire le surcoût relatif en termes de surface de silicium. Pour cette première étude, nous prendrons les hypothèses suivantes :

- Les données sont stockées dans une table
- Les données sont ordonnées dans cette table grâce à la relation de cohérence. De cette façon, on obtient une seule période du signal échantillonné
- Seules les premières harmoniques seront estimées.

Trois méthodes sont étudiées : la transformée de Fourier Rapide, l'algorithme de Goertzel, et une nouvelle méthode intitulée LoCHE, pour Low Cost Harmonics Estimation. Chacune de ces méthodes est exposée ci-dessous, et fait l'objet d'une implémentation sur FPGA (Altera Stratix II). Le choix de ces trois méthodes nous est apparu légitime au regard de la littérature vaste et variée. La méthode classique de Fourier représente la méthode incontournable de l'analyse spectrale : bien qu'il existe de nombreuses façons de l'implémenter, nous nous contenterons d'étudier l'algorithme originel dit radix-2, basé sur l'opérateur papillon de première forme. L'algorithme de Goertzel représente quant à lui une manière directe et simplifiée de calculer un nombre de raies peu élevé. Enfin, nous proposerons un nouvel algorithme basé sur une décomposition du signal sinusoïdal en série de Fourier, et dont le calcul des paramètres ne requiert principalement que des additionneurs et soustracteurs [FBD⁺06].

5.3.1 Notations

Afin de conserver une cohérence face aux nombreuses façons de présenter l'analyse spectrale, nous établissons tout d'abord une base de notations des différentes variables :

$f(t)$ représente le signal temporel.

F_{in} représente la fréquence du signal d'entrée.

T représente la période du signal d'entrée.

f_{ech} représente la fréquence d'échantillonnage.

T_{ech} représente la période d'échantillonnage.

N représente le nombre d'échantillons.

f_j représente le signal échantillonné, $j \in [0..N-1]$, $f_j = f(j.T_{ech})$.

M représente le nombre de cycles, c'est un entier impair qui répond à la relation de cohérence suivante :

$$N \cdot F_{in} = M \cdot f_{ech}.$$

W_N représente le nombre complexe défini par $W_N = e^{\frac{2i\pi}{N}}$.

F_k représente la transformée de Fourier discrète du signal f , $k \in [0..N-1]$.

5.3.2 La transformée de Fourier

En 1822, Joseph Fourier expose dans son traité *Théorie analytique de la chaleur* [Fou22], les séries et la transformation qui porteront son nom. Il énonce qu'une fonction périodique peut être décomposée sous forme de série trigonométrique et fournit alors les bases de l'analyse harmonique.

La décomposition en séries de Fourier énonce que tout signal T-périodique peut être décrit par une somme infinie de fonctions trigonométriques. Dans le chapitre 3, ces séries ont déjà été présentées. Dans le cadre d'un signal échantillonné, la transformée de Fourier discrète du signal f , calculée sur N échantillons, vaut pour la raie d'ordre k :

$$F_N(k) = \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2i\pi k j}{N}} = \sum_{j=0}^{N-1} f_j \cdot W_N^{kj} \quad (5.6)$$

Pendant de nombreuses années, la façon standard de calculer le nombre complexe $F_N(\cdot)$ était d'effectuer l'ensemble des calculs décrits dans l'équation précédente. Cet ensemble d'opérations fournit une complexité de calcul d'ordre $O(N^2)$. Dans les années 60, J.W. Cooley et J.W. Tukey généralisèrent le calcul de $F_N(\cdot)$ au travers d'un algorithme nommé *Fast Fourier Transform* (FFT) dont la complexité s'avère être de l'ordre de $\log_2(N)O(N)$. Leurs travaux peuvent être expliqués à partir de la théorie de décomposition exposé en 1942 par Danielson et Lanczos [Dan40]. Le lecteur trouvera en annexe une démonstration de cette décomposition.

Cette méthode permet d'obtenir la totalité des transformées de Fourier $F_N(\cdot)$, or la section précédente nous indique que seules quelques raies sont suffisantes pour extraire les non-linéarités. L'algorithme produit plus de résultats que nécessaire, il est donc possible de le simplifier afin de n'utiliser que les calculs des raies désirées. La figure 5.14 représente le "chemin de calcul" de la transformée $F_8(2)$.

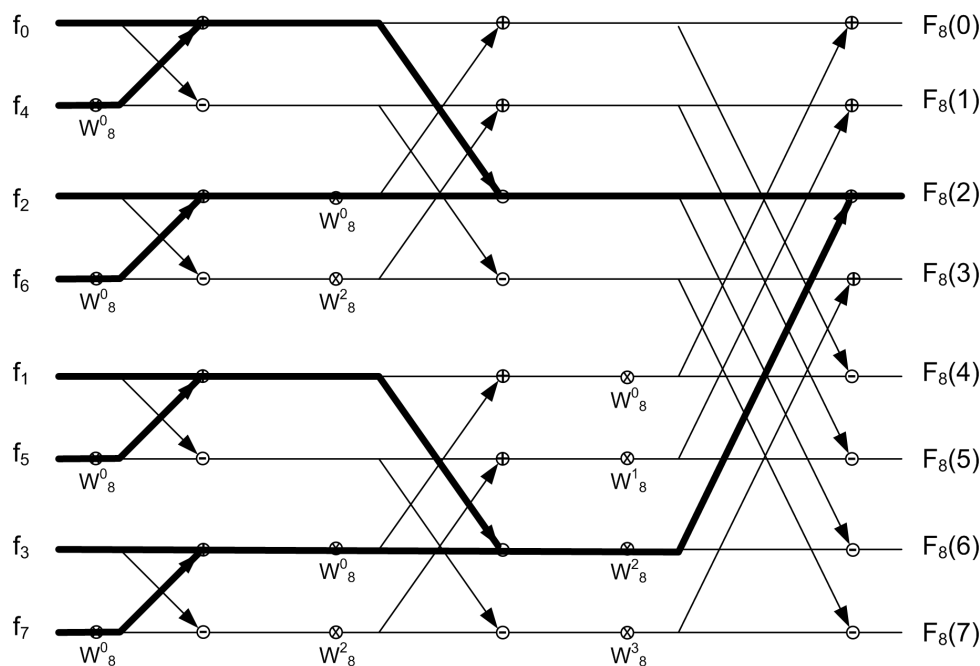
La traduction de ce "chemin de calcul" se traduit par l'équation suivante.

$$F_8(2) = ((f_0 + W_8^0 \cdot f_4) - W_8^0 \cdot (f_2 + W_8^0 \cdot f_6)) + W_8^2 \cdot ((f_1 + W_8^0 \cdot f_5) - W_8^0 \cdot (f_3 + W_8^0 \cdot f_7)) \quad (5.7)$$

Ce chemin de calcul réduit la complexité du calcul du fait qu'étage après étage, une seule des deux sorties de l'opérateur papillon est utilisée. L'utilisation des nombres complexes W_N est également limité. Dans le cas où $N = 2^{12} = 4096$, et que seules les 6 premières harmoniques sont désirées, le calcul des 6 chemins ne réclame que 31 valeurs de W_N au lieu des $2^{11} = 2048$ valeurs originelles. Ce gain n'est pas négligeable dans notre contexte de calcul embarqué.

L'utilisation de cette transformée de Fourier réduite nous permet de diminuer les ressources nécessaires au calcul. Afin de quantifier ces ressources, nous avons comparé trois implémentations différentes :

- l'implémentation classique issue du fournisseur de FPGA : Altera,

FIG. 5.14 – Calcul de la transformée $F_8(2)$

- l'implémentation classique issue du fournisseur libre OpenCore.org,
- l'implémentation séquentielle optimisée pour nos besoins.

Dans tout les cas, le circuit est synthétisé de façon à utiliser l'ensemble des ressources du FPGA : les éléments logiques configurables et les blocs arithmétiques dédiés (éléments DSP [Alt05]). Les résultats de l'occupation de ces ressources de base sont présentés dans le tableau 5.2. On observe effectivement que l'algorithme séquentiel utilise beaucoup moins de ressources que les deux autres. La grande différence provient du fait que dans les deux autres algorithmes, la plupart des calculs se fait de manière parallèle, alors que notre implémentation séquentielle réutilise les mêmes blocs. La durée de calcul dans ce dernier cas est beaucoup plus longue. Il faudrait optimiser les implémentations parallèles de façon à réutiliser également les éléments internes. Le nombre de cellules logiques devrait alors diminuer, et la comparaison serait plus pertinente. Ces implémentations n'ont pas été réalisées dans cette étude.

Algorithme	Cellules Logiques	éléments DSP
Altera	49851	96
OpenCore.org	14794	96
Séquentiel	403	92

TAB. 5.2 – Comparaison des ressources utilisées au sein du FPGA

5.3.3 L'algorithme de Goertzel

L'algorithme de Goertzel a été présenté pour la première fois par Gerald Goertzel en 1958 [Goe58]. Cet algorithme propose une solution à faible complexité de la détermination de la valeur de quelques raies

spectrales. Là où l'algorithme de FFT fourni le spectre total du signal, Goertzel fournit seulement la valeur des raies désirées. Comme indiqué dans la section précédente, cette méthode répond parfaitement à nos besoins. Une des applications de cet algorithme est la détection de signaux aux fréquences prédéterminées comme c'est le cas dans les claviers téléphoniques utilisant le standard DTMF (*Dual-tone multi-frequency*).

L'algorithme de Goertzel effectue le calcul suivant à partir des N échantillons (avec $f[-1] = f[-2] = 0$) :

$$f_{g1}[p] = f[p] + 2\cos\left(2\pi\frac{k}{N}\right)f[n-1] - f[n-2] \text{ pour } p \in [0..N[\quad (5.8)$$

Une seconde équation est appliquée à la sortie du premier calcul :

$$f_{g2}[p] = f_{g1}[p] - e^{-2j\pi\frac{k}{N}} \times f_{g1}[p-1] \quad (5.9)$$

Ces deux opérations constituent l'application de deux filtres à réponse impulsionnelle infinie. Leurs traductions en transformée en z s'exprime de la façon suivante :

$$\frac{F_{g1}(z)}{F(z)} = \frac{1}{1 - 2\cos\left(2\pi\frac{k}{N}\right)z^{-1} + z^{-2}} = \frac{1}{(1 - e^{+2\pi i\frac{k}{N}}z^{-1})(1 - e^{-2\pi i\frac{k}{N}}z^{-1})} \quad (5.10)$$

et

$$\frac{F_{g2}(z)}{F_{g1}(z)} = 1 - e^{-2\pi i\frac{k}{N}}z^{-1} \quad (5.11)$$

La mise en cascade de ces deux filtres fournit une représentation sous forme de filtre de l'algorithme de Goertzel.

$$\frac{F_{g1}(z) F_{g2}(z)}{F(z) F_{g1}(z)} = \frac{F_{g2}(z)}{F(z)} = \frac{(1 - e^{-2\pi i\frac{k}{N}}z^{-1})}{(1 - e^{+2\pi i\frac{k}{N}}z^{-1})(1 - e^{-2\pi i\frac{k}{N}}z^{-1})} = \frac{1}{1 - e^{+2\pi i\frac{k}{N}}z^{-1}} \quad (5.12)$$

En revenant dans le domaine temporel, cette fonction de transfert se traduit par les équations suivantes :

$$f_{g2}[p] = f[p] + e^{2\pi i\frac{k}{N}}f_{g2}[p-1] = \sum_{q=-\infty}^p f[q] e^{2\pi i\frac{k}{N}(p-q)} = e^{2\pi i\frac{k}{N}p} \sum_{q=-\infty}^p f[q] e^{-2\pi i\frac{k}{N}q} \quad (5.13)$$

or pour $q < 0$, $f[q] = 0$. Donc :

$$f_{g2}[p] = e^{2\pi i\frac{k}{N}p} \sum_{q=0}^p f[q] e^{-2\pi i\frac{k}{N}q} \quad (5.14)$$

Pour $p < N$, la valeur représentée par $f_{g2}[p]$ n'a aucune signification pertinente. Par contre, pour $p = N$, et en posant $f[N] = 0$, on obtient la valeur de la transformée de Fourier discrète du signal f :

$$f_{g2}[N] = e^{2\pi i\frac{k}{N}N} \sum_{q=0}^{N-1} f[q] e^{-2\pi i\frac{k}{N}q} = F_N(k) \quad (5.15)$$

La figure 5.15 représente l'architecture du filtre de Goertzel.

L'algorithme nécessite $N + 2$ multiplications réelles et $2 \times N + 1$ additions réelles. En comparant ces valeurs à celles nécessaires pour l'algorithme de FFT, l'algorithme de Goertzel reste pertinent tant qu'on se

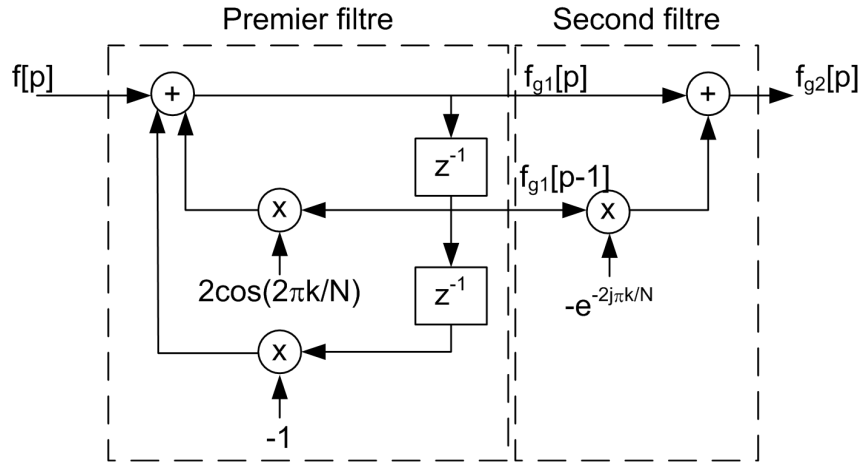


FIG. 5.15 – Filtre RII de Goertzel

limite à une dizaine de raies à calculer. La précision de l’algorithme est identique à celle de l’algorithme de FFT lorsque l’on utilise le même nombre d’échantillons. Le nombre encore élevé de multiplications, et le calcul effectué à partir de nombre de type réel constitue également un coût important au niveau ressource dans le cas d’une implémentation embarquée. Il est possible de diminuer ces coûts en manipulant différemment le signal converti. C’est l’objet de notre proposition d’algorithme présentée dans la section suivante.

5.3.4 Notre solution : l’algorithme LoCHE

L’algorithme précédent a permis la réduction de la complexité des calculs, mais le nombre de multiplications est encore élevé, et le type des données traitées n’étant pas du type entier, il réclame une représentation sur un nombre élevé de bits. Ces considérations seront exposées dans la dernière section de ce chapitre. Dans cette section nous présentons un nouvel algorithme d’extraction spectrale, basé sur une manipulation additive ou soustractive des données de type entier.

Cet algorithme nommé LoCHE, pour LOW Cost Harmonics Estimation [FBD⁺06] a été mis au point sous l’influence de la lecture de [SN97] écrit par Sunter. Celui-ci tente de retrouver une écriture polynomiale d’une fonction de transfert à partir de manipulations additives et soustractives de sommes intégrales de la courbe représentant cette fonction. Notre algorithme reprend ce concept, mais en utilisant les séries de Fourier en lieu et place du modèle polynomial. En effet, nous cherchons à modéliser un signal sinusoïdal et non plus une fonction de type linéaire.

Objectif de notre méthode

Dans le chapitre 3, il a été démontré que toute fonction périodique pouvait se décomposer en une série de Fourier. Le signal d’entrée étant de type T-périodique, cette décomposition peut donc s’appliquer.

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos(k\omega t) + b_k \sin(k\omega t)) \quad (5.16)$$

avec $\omega = \frac{2\pi}{T}$. Dans la pratique, l'ordre de la modélisation est limité à la valeur H_{MAX} . Les paramètres recherchés sont les valeurs des couples (a_k, b_k) . Dans un premier temps, nous utilisons deux nouvelles fonctions, $f_c(\cdot)$ et $f_s(\cdot)$ afin de séparer la recherche des valeurs a_k de celle des valeurs b_k . Ces deux nouvelles fonctions utilisent les propriétés de symétrie des fonctions *cosinus* et *sinus*.

$$f_c(t) = f(t) + f(-t) \approx 2 \sum_{k=1}^{H_{MAX}} (a_k \cos(k\omega t)) \quad (5.17)$$

$$f_s(t) = f(t) - f(-t) \approx 2 \sum_{k=1}^{H_{MAX}} (b_k \sin(k\omega t)) \quad (5.18)$$

Dans le domaine discret, ces deux fonctions s'expriment facilement, si l'on considère un nombre N d'échantillons.

$$f_c[n] = f\left[\frac{N}{2} - n + 1\right] + f[n] \quad (5.19)$$

$$f_s[n] = f\left[\frac{N}{2} - n + 1\right] - f[n] \quad \text{avec } n \in \left[1.. \frac{N}{2}\right] \quad (5.20)$$

A partir de ces nouvelles fonctions, les recherches des valeurs a_k et b_k peuvent être réalisées de manière indépendante. La figure 5.16 représente une fonction f quelconque, et la figure 5.17 illustre la décomposition de f en f_c et f_s .

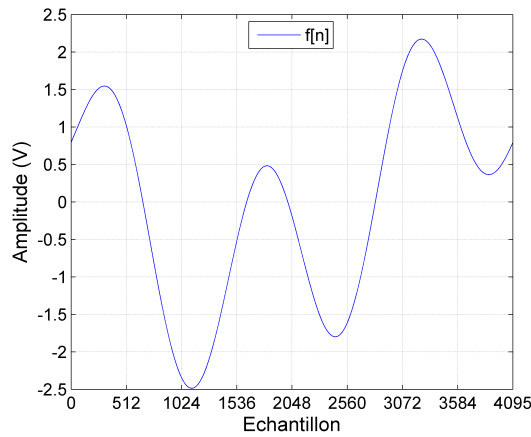


FIG. 5.16 – Fonction $f[n]$

Démonstration sur un modèle d'ordre 3

Dans cet exemple, nous considérerons un modèle d'ordre $H_{MAX} = 3$, et nous ne nous intéresserons qu'à la recherche des valeurs de a_k , le mode de recherche des valeurs de b_k s'effectuant de manière similaire. Nous ne travaillerons donc qu'avec la fonction f_c , et les paramètres a_1, a_2 et a_3 . La fonction f_c ainsi que

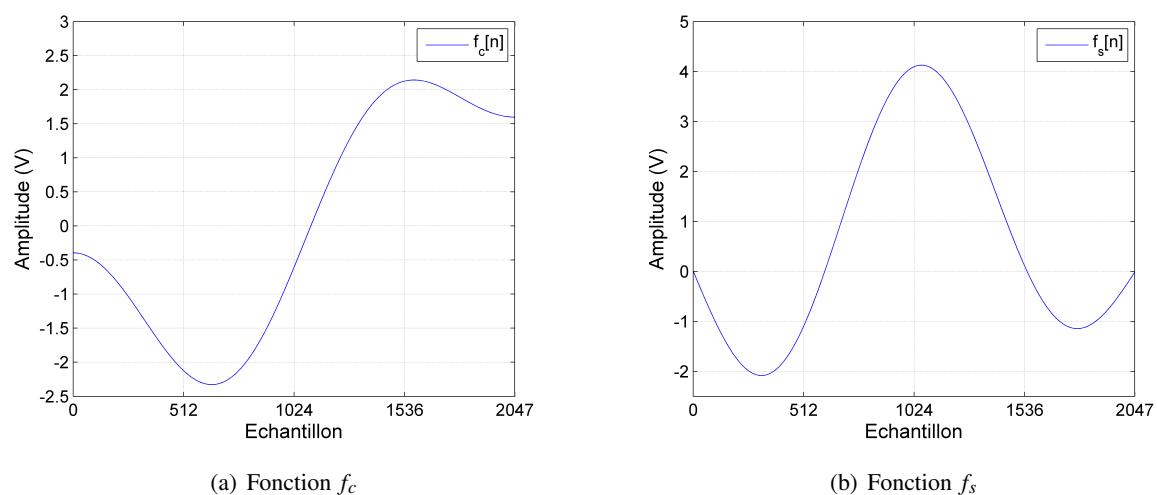


FIG. 5.17 – Décomposition de la fonction $f[n]$

sa décomposition en cosinus sont représentées sur la figure 5.18. On cherche à discriminer d'une façon simple chaque amplitude des trois cosinus de la décomposition.

$$f_c(t) \approx 2 \sum_{k=1}^3 (a_k \cos(k\omega t)) \quad (5.21)$$

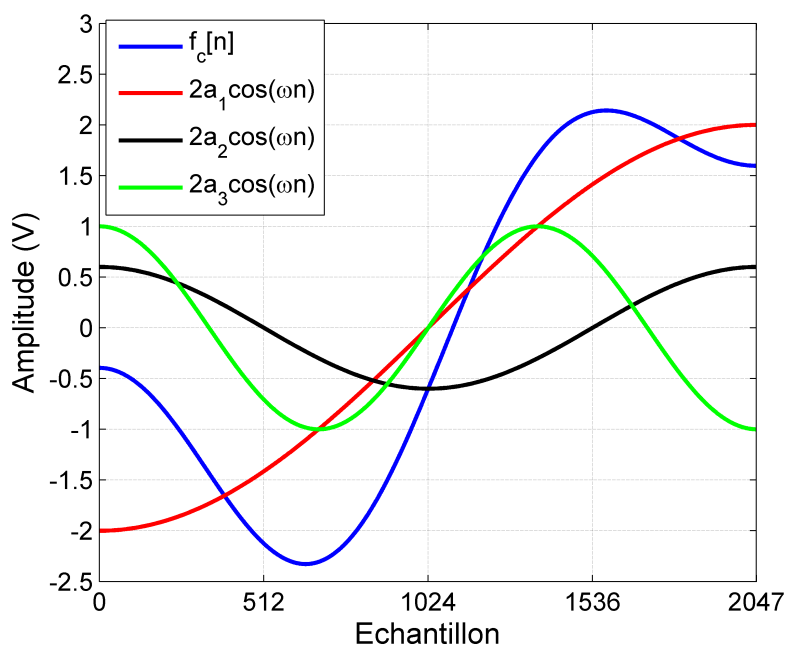


FIG. 5.18 – Décomposition du modèle d'ordre 3

Si l'on multiplie la fonction f_c et ses composantes internes, par la fonction $\text{signe}(\cos(2\omega t))$, la figure

5.19 est obtenue. La fonction $signe(\cdot)$ est définie ci-dessous :

$$signe(x) = \begin{cases} +1 & \text{si } x > 0, \\ -1 & \text{si } x < 0. \end{cases} \quad (5.22)$$

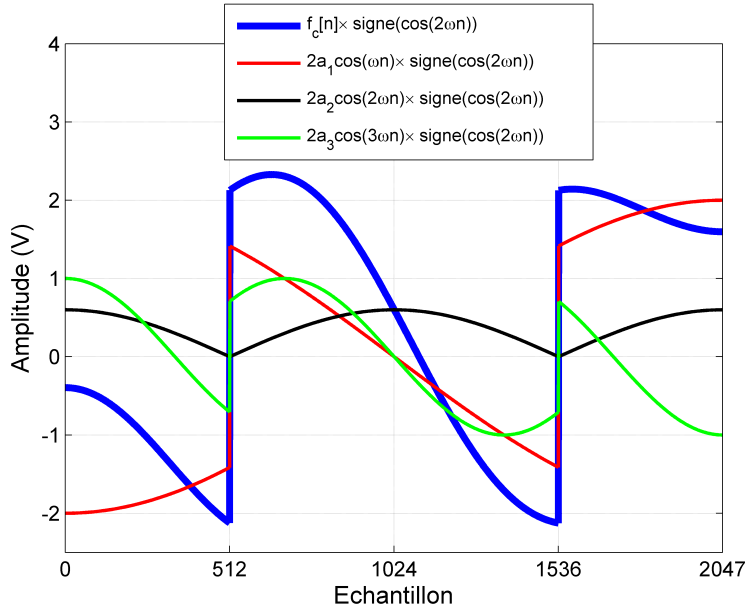


FIG. 5.19 – Décomposition du modèle d'ordre 3

Graphiquement, on peut observer que le calcul intégral des différentes courbes sur l'ensemble de l'axe des abscisses semble fournir des résultats intéressants. En effet, les intégrales de $2a_1 \cos(\omega n) \times signe(\cos(2\omega n))$ et $2a_3 \cos(3\omega n) \times signe(\cos(2\omega n))$ sont nulles, alors que l'intégrale de $2a_2 \cos(2\omega n) \times signe(\cos(2\omega n))$ ne l'est pas. Or si les contributions des cosinus $\cos(\omega n)$ et $\cos(3\omega n)$ sont nulles, cela revient à dire que l'intégrale de $f_c(n) \times signe(\cos(2\omega n))$ est égale à celle de $2a_2 \cos(2\omega n) \times signe(\cos(2\omega n))$. A partir de f_c on est donc à même de déterminer a_2 . Les calculs suivants démontrent cette observation.

$$\begin{aligned} F_c^2 &= \int_0^{\frac{N}{2} \times T_{ech}} f_c(t) signe(\cos(2\omega t)) dt \\ &= \int_0^{\frac{N}{2} \times T_{ech}} 2 \sum_{k=1}^3 (a_k \cos(k\omega t)) signe(\cos(2\omega t)) dt \\ &= \sum_{k=1}^3 \left(2 \cdot a_k \int_0^{\frac{N}{2} \times T_{ech}} \cos(k\omega t) signe(\cos(2\omega t)) dt \right) \\ &= 2 \cdot a_1 \cdot 0 + 2 \cdot a_2 \cdot \frac{2}{\omega} + 2 \cdot a_3 \cdot 0 = a_2 \cdot \frac{4}{\omega} \end{aligned} \quad (5.23)$$

On obtient donc la relation directe qui relie a_2 au calcul de l'intégrale F_c^2 de $f_c(t) signe(\cos(2\omega t))$.

$$a_2 = \frac{\omega}{4} \cdot \int_0^{\frac{N}{2} \times T_{ech}} f_c(t) \cdot signe(\cos(2\omega t)) dt \quad (5.24)$$

5.3. CALCUL DES HARMONIQUES IN SITU

Ce qui se traduit dans le cas discret par la relation suivante :

$$a_2 = \frac{\omega}{4} \left(\sum_0^{\frac{N}{8}} f_c[n] - \sum_{\frac{N}{8}}^{\frac{3N}{8}} f_c[n] + \sum_{\frac{3N}{8}}^{\frac{N}{2}} f_c[n] \right) \quad (5.25)$$

En utilisant la fonction *signe* ($\cos(3\omega t)$), on met en évidence la relation qui lie a_3 et le calcul intégral F_c^3 .

$$a_3 = F_c^3 = \frac{\omega}{4} \left(\sum_0^{\frac{N}{12}} f_c[n] - \sum_{\frac{N}{12}}^{\frac{3N}{12}} f_c[n] + \sum_{\frac{3N}{12}}^{\frac{5N}{12}} f_c[n] - \sum_{\frac{5N}{12}}^{\frac{N}{2}} f_c[n] \right) \quad (5.26)$$

Dans le cas du calcul de a_1 , le calcul est un peu moins trivial, et sera expliqué dans la section suivante. Le calcul de a_1 tient compte du calcul au préalable de a_3 . On obtient l'expression suivante :

$$a_1 = \frac{\omega}{4} \left(\sum_0^{\frac{N}{4}} f_c[n] - \sum_{\frac{N}{4}}^{\frac{N}{2}} f_c[n] \right) - \frac{1}{6} a_3 \quad (5.27)$$

Ces calculs mettent en évidence qu'à partir de simples additions et soustractions effectuées sur le signal discrétisé, les paramètres a_1 , a_2 et a_3 peuvent être extraits facilement. Pour les valeurs de b_k , les calculs se font de manière similaire en utilisant les fonctions *signe* ($\sin(k\omega t)$). La section suivante expose la généralisation de la méthode pour un ordre H_{MAX} quelconque.

Généralisation de la méthode

Dans le cas de la modélisation à l'ordre H_{MAX} , seule les valeurs de a_k seront également exprimées. Les valeurs de b_k suivent le même type de raisonnement. Le calcul des intégrales du type F_c^q s'exprime de la façon suivante :

$$F_c^q = \int_0^{\frac{NT_{ech}}{2}} f_c(t) \cdot \text{signe}(\cos(q\omega t)) dt \quad (5.28)$$

Ce qui se traduit grâce à la décomposition en série de Fourier en :

$$\begin{aligned} F_c^q &= \int_0^{\frac{NT_{ech}}{2}} \left(2 \sum_{k=1}^{H_{MAX}} (a_k \cos(k\omega t)) \right) \text{signe}(\cos(q\omega t)) dt \\ &= 2 \sum_{k=1}^{H_{MAX}} \left(a_k \int_0^{\frac{NT_{ech}}{2}} \cos(k\omega t) \cdot \text{signe}(\cos(q\omega t)) dt \right) \end{aligned} \quad (5.29)$$

On note C_{qk} la contribution du cosinus $\cos(k\omega x)$ lorsque l'on cherche à calculer la valeur du paramètre a_q :

$$C_{qk} = \int_0^{\frac{NT_{ech}}{2}} \cos(k\omega t) \cdot \text{signe}(\cos(q\omega t)) dt \quad (5.30)$$

On obtient alors la nouvelle expression de F_c^q :

$$F_c^q = 2 \sum_{k=1}^{H_{MAX}} (a_k C_{qk}) \quad (5.31)$$

En faisant varier la valeur de q , on peut écrire notre problème sous forme matricielle :

$$F_c = \begin{bmatrix} F_c^1 \\ \vdots \\ F_c^k \\ \vdots \\ F_c^{H_{MAX}} \end{bmatrix} = 2 \cdot \begin{bmatrix} C_{11} & \dots & C_{1k} & \dots & C_{1H_{MAX}} \\ \vdots & & \vdots & & \vdots \\ C_{q1} & \dots & C_{qk} & \dots & C_{qH_{MAX}} \\ \vdots & & \vdots & & \vdots \\ C_{H_{MAX}1} & \dots & C_{H_{MAX}k} & \dots & C_{H_{MAX}H_{MAX}} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_{H_{MAX}} \end{bmatrix} = 2CA \quad (5.32)$$

L'expression $\text{signe}(\cos(q\omega t))$ ne peut prendre que 2 valeur (1 ou -1), et ces valeurs sont constantes sur les intervalles délimités par deux zéros consécutifs de la fonction $\cos(q\omega t)$. C_{qk} peut donc être évalué de façon simple en tenant compte des zéros de $\cos(q\omega t)$ dans l'intervalle $[0, \frac{NT_{ech}}{2}]$. Soient $(z_i)_{i \in [1..q]}$ les zéros précités, alors en rajoutant $z_0 = 0$ et $z_{q+1} = \frac{NT_{ech}}{2}$ pour les valeurs extrêmes, C_{qk} peut s'exprimer de la manière suivante :

$$C_{qk} = \sum_{i=0}^q (-1)^i \int_{z_i}^{z_{i+1}} \cos(k\omega t) dt \quad (5.33)$$

On peut alors démontrer les relations suivantes :

– Si $k = q$ alors

$$C_{qq} = \frac{2}{\omega} \quad (5.34)$$

– Si q est impair ($q = 2p + 1$) et si k est impair ($k = 2t + 1$) et si $\frac{k}{q}$ est un entier impair, alors :

$$C_{qk} = (-1)^{p+t} \cdot \frac{2q}{k\omega} \quad (5.35)$$

– Si q est pair ($q = 2p$) et si k est pair ($k = 2t$) et si $\frac{k}{q}$ est un entier impair ($k/q = 2m + 1$), alors :

$$C_{qk} = (-1)^{p+t+m} \cdot \frac{2q}{k\omega} \quad (5.36)$$

– Sinon

$$C_{qk} = 0 \quad (5.37)$$

Ces relations fortes, font que la matrice C est majoritairement constituée de 0 et possède de plus, la propriété d'être une matrice triangulaire supérieure. Elle est donc aisément inversible à l'aide de la méthode de Gauss-Jordan. Les valeurs des a_k peuvent donc s'exprimer très simplement :

$$a_{H_{MAX}} = \frac{\omega F_c^{H_{MAX}}}{4} \quad (5.38)$$

Pour $i \in \{1, \dots, H_{MAX} - 1\}$

– Si i est pair, alors :

$$a_i = \frac{\omega}{4} \left(F_c^i - \sum_{m=1}^{m \leq \frac{1}{2}(H_{MAX} - i)} (-1)^{(m+1)i+m} \frac{2}{(2m+1)\omega} a_{(2m+1)i} \right) \quad (5.39)$$

– Si i est impair :

$$a_i = \frac{\omega}{4} \left(F_c^i - \sum_{m=1}^{m \leq \frac{1}{2} \left(\frac{H_{MAX}}{i} - 1 \right)} (-1)^{(m+1)i-1} \frac{2}{(2m+1)\omega} a_{(2m+1)i} \right) \quad (5.40)$$

De façon similaire, on utilise les intégrales F_s^q pour calculer les valeurs de b_k :

$$F_s^q = \int_0^{\frac{NT_{ech}}{2}} f_s(t) \cdot \text{signe}(\sin(q\omega t)) dt \quad (5.41)$$

Leur valeur est alors fournie par les équations suivantes :

$$b_{H_{MAX}} = \frac{\omega \cdot F_s^{H_{MAX}}}{4} \quad (5.42)$$

Pour $i \in \{1, \dots, H_{MAX} - 1\}$

$$b_i = \left(\frac{\omega}{4} F_s^i - \sum_{m=1}^{m \leq \frac{1}{2} \left(\frac{H_{MAX}}{i} - 1 \right)} \frac{1}{2(2m+1)} b_{(2m+1)i} \right) \quad (5.43)$$

Ces équations peuvent paraître compliquées de prime abord, mais leur utilisation n'intervient qu'à la fin du calcul. La partie la plus longue de notre algorithme consiste à estimer les intégrales F_c^q et F_s^q . La complexité globale de l'algorithme est très faible : pour le calcul d'une raie, environ N additions/soustractions sont nécessaires pour le calcul des intégrales, et quelques multiplications pour normaliser le résultat final. La section suivante compare les implémentations physiques des différents algorithmes.

5.3.5 Implémentation et comparaison des techniques

Les trois algorithmes précédents ont été décrits en VHDL, synthétisés et implantés au sein de notre véhicule de test. Nous ne nous attarderons pas sur les différents codes écrits, mais nous nous contenterons d'illustrer leur fonctionnement au travers des figures 5.20 à 5.22. L'algorithme de FFT simplifié a été suffisamment représenté dans la section précédente, aussi, aucune figure ne reprendra son explication.

Algorithme de Goertzel

L'algorithme de Goertzel a été implémenté sous forme de bloc (figure 5.20) reprenant l'écriture sous forme de filtre RII de la figure 5.15. La représentation des données est ici indiquée par les termes entre crochets. Par exemple, $Data_in[11.0]$ illustre une représentation sur 12bits de la donnée $Data_in$.

$Data_in$ constitue l'entrée du bloc VHDL : sa représentation sur 12bits provient de la résolution du convertisseur lié à notre étude. Comme l'algorithme utilise des nombres signés, la première opération effectuée est une extension du bit de signe ainsi qu'une augmentation de la représentation de 12 à 36bits (= [24.12]). Cette augmentation est nécessaire pour assurer une précision suffisante lors des calculs intermédiaires. Le résultat de la multiplication n'est pas représenté sur $36 + 32$ bits car la constante est en fait un nombre réel, représenté sur 2bits pour sa partie entière et sur 30bits pour sa partie décimale. La

multiplication procède également à un arrondi afin d'obtenir au final un nombre représenté sur 36bits. L'opération de "complément à deux" permet de procéder à la multiplication par -1 nécessaire. Enfin, l'additionneur à trois entrées conserve la représentation sur 36bits, cette dernière ayant été choisie afin de ne jamais saturer.

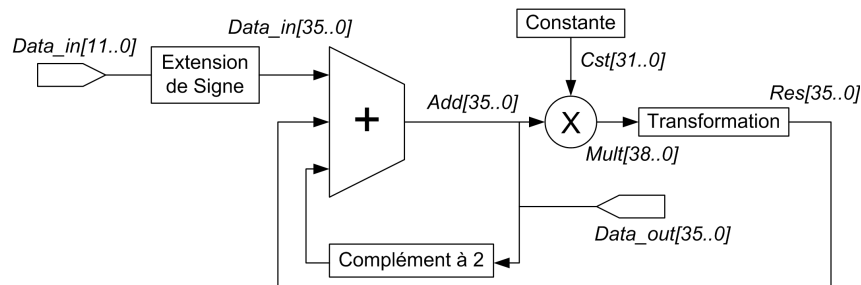


FIG. 5.20 – Architecture de l'implantation de l'algorithme de Goertzel

Ce bloc de base est ensuite répété 5 fois (figure 5.21) et ajusté pour que la valeur de la constante soit adaptée à la raie recherchée : $Constante_k = 2\cos(2\pi k/N)$ pour $k = \{1, 2, 3, 4, 5\}$. Les données issues du convertisseur ont été préalablement stockées dans une mémoire de taille $N \times 12bits$. Une machine d'état contrôle ensuite la valeur de l'adresse à utiliser et active le calcul parallèle des différents blocs. Tous les éléments fonctionnent de façon synchrone. Le résultat des calculs est ensuite accessible via les registres de données nommés *Raie_k*.

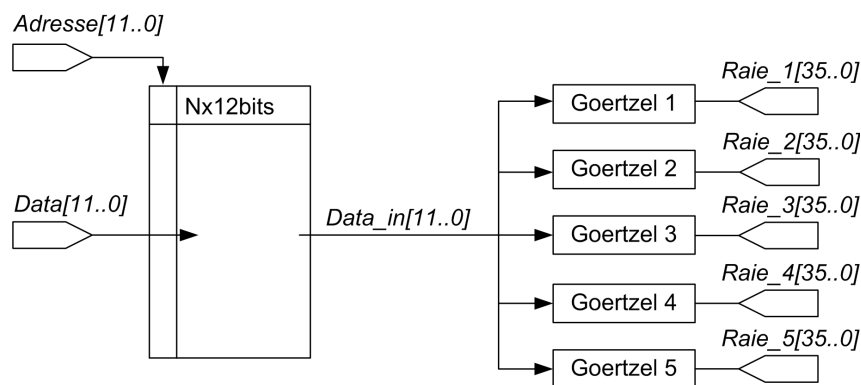


FIG. 5.21 – Architecture du circuit de calcul des 5 premières raies du spectre

Algorithme LoCHE

L'algorithme LoCHE n'utilise que des additions et des soustractions pour calculer les valeurs des F_c^k et F_s^k . Pour effectuer cette opération, un opérateur *AddSub* est utilisé (figure 5.22). Cet opérateur est contrôlable à partir d'un simple bit, et permet d'agir soit en additionneur, soit en soustracteur. La valeur de ce bit de contrôle est fourni par la mémoire servant à stocker les signes des 10 opérateurs *AddSub* (5 pour le calcul des F_c^k et 5 pour le calcul des F_s^k , avec $k = \{1, 2, 3, 4, 5\}$). Ce stockage peut être remplacé

5.3. CALCUL DES HARMONIQUES IN SITU

par un calcul à la volée des bits de contrôle de signe. Son implémentation n'a pas encore été réalisée, mais peut s'effectuer à l'aide de simples compteurs. Le bouclage de la sortie des opérateurs *AddSub* permet de les faire fonctionner comme un accumulateur. Afin de ne pas saturer les valeurs, la représentation des données s'effectue sur 24bits, l'algorithme ne travaillant uniquement qu'avec des nombres entiers. De la même manière que pour l'algorithme de Goertzel, une machine d'état contrôle l'envoi des données stockées dans la table mémoire. Toutes les opérations sont également traitées de façon synchrone. Les blocs combinatoires *C* et *S* représentent les opérations linéaires à appliquer aux variables F_c^k et F_s^k afin de calculer la valeur des paramètres a_k et b_k . Ces combinaisons peuvent être éventuellement effectuées en externe, les variables de sorties F_c^k et F_s^k représentant alors une signature des valeurs des harmoniques du signal.

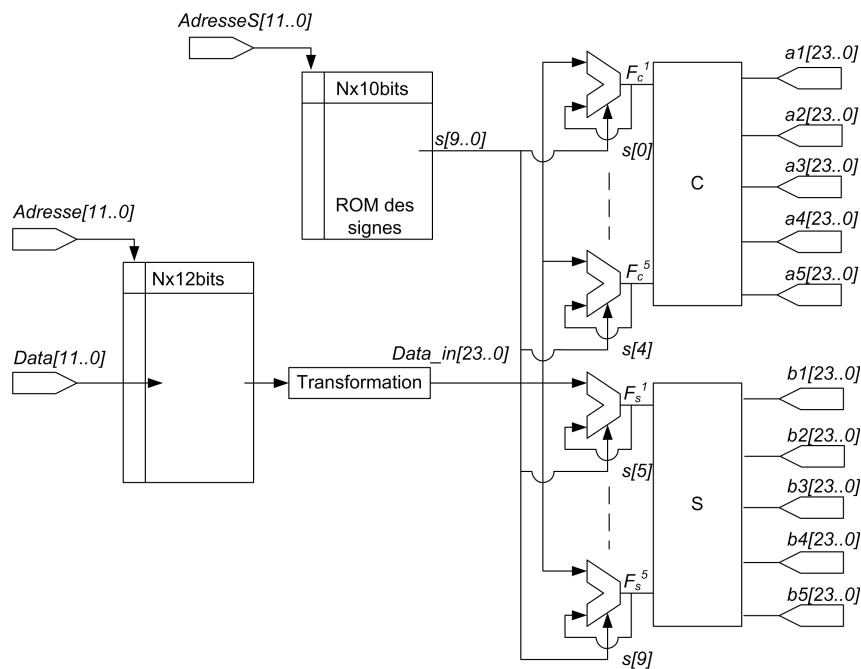


FIG. 5.22 – Architecture du circuit de calcul des 5 premières raies du spectre à partir de l'algorithme LOCHE

Comparaison des implémentations

Deux éléments sont pris en compte dans notre étude comparative des algorithmes d'estimation spectrale :

- les ressources nécessaires aux calculs (taille mémoire, nombre de portes)
- la qualité de l'estimation.

Le tableau 5.3 synthétise les ressources nécessaires lors de nos diverses implémentations. Les *Logic Elements* représentent la plus petite entité présente dans le FPGA, aussi, leur nombre est assimilable au nombre de portes nécessaire. Les blocs DSP représentent une entité de calcul propre au FPGA. Leur conception interne utilise des ressources optimisées pour le calcul : portes, mémoire, registres. Ils sont notamment utilisés pour le calcul d'opérations du type multiplication. Dans le FPGA choisi dans cette étude [Alt05], le nombre de blocs DSP est 96. Ces blocs représentent une surface de silicium considérable

au sein du composant, aussi, plus le nombre de blocs utilisé est faible, moins la surface utilisée sera grande. Le nombre de bits mémoire RAM représente les mémoires utilisées pour l'extraction. Il est à noter que dans nos implémentations, toutes les données issues du convertisseur étaient stockées dans une mémoire de type RAM. La valeur prend donc en compte ce paramètre. Enfin, la fréquence F_{max} représente la fréquence maximale de fonctionnement de l'implémentation physique.

TAB. 5.3 – Ressources internes utilisées dans le FPGA

	Goertzel	LoCHE	FFT Altera	FFT OpenCore.org	FFT séquentielle
# Logic Elements	3,796	1,080	49,851	14,794	403
# blocs DSP	0	0	96	96	92
# RAM bits	49,152	90,112	49,512	49,512	90,112
F_{max} (MHz)	63.07	144.99	300	77.71	46.42

Il ressort de ce tableau que ce sont les algorithmes de FFT qui consomment le plus de ressources, notamment en terme de blocs DSP utilisés. Ces algorithmes sont donc difficilement implémentables pour une extraction spectrale embarquée. Les deux algorithmes qui semblent les plus adaptés aux contraintes de l'embarqué, sont ceux de Goertzel et LoCHE. Leur utilisation de Logic Elements est faible et ils n'utilisent aucun blocs DSP. Le nombre élevé de bits utilisés pour LoCHE tient compte de la mémoire des signes des opérateurs *AddSub*, or il a été mentionné auparavant qu'une nouvelle implémentation à l'aide de compteur devrait permettre de nous affranchir de cette contrainte. La fréquence maximale permet de trancher entre les deux algorithmes. En effet, l'algorithme de Goertzel, du fait des multiplications utilisées, ne peut fonctionner aussi vite que l'algorithme LoCHE. La fréquence maximale atteinte par LoCHE étant supérieure aux fréquences d'utilisation du convertisseur étudié, lui permettrait de traiter les données à la volée, et donc de ne pas utiliser un stockage des données.

D'un point de vue qualitatif, la figure 5.23 compare les estimations des différentes méthodes. Les résultats fournis par la FFT et par Goertzel sont identiques, à l'exception de la raie numéro 4, dont le niveau de puissance se situe très proche du plancher de bruit dû à la quantification. L'algorithme LoCHE réussit à estimer avec autant de précision les 3 premières raies, mais est plus sensible au niveau faible de la puissance de la 4ème harmonique. L'estimation de la 5ème raie est également erronée. La justification de cette erreur provient du calcul des arrondis des zéros z_i mentionnés dans l'équation 5.33.

5.3.6 Conclusion

Le niveau de précision et la faible quantité de ressources nécessaires aux algorithmes de Goertzel et LoCHE en font deux candidats à l'extraction spectrale embarquée. Cependant, le nombre de raies harmoniques calculées est trop faible pour pouvoir utiliser leur valeur dans un processus de compensation embarquée. Leur utilisation s'avère par contre intéressante dans un contexte de test in situ de type BIST (*Built-In Self Test*).

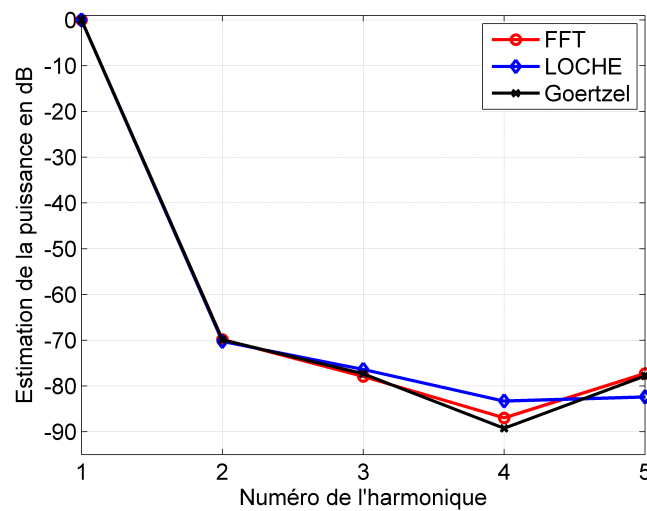


FIG. 5.23 – Comparaison des estimations

5.4 Conclusion

Ce chapitre nous a permis de mieux comprendre les besoins d'un système de compensation embarqué. Dans un premier temps nous avons mis en oeuvre une simplification de la méthode de compensation liée à une étude de la courbe de non-linéarités propre au convertisseur. Nous avons pu mettre en avant de nombreuses simplifications liées à une compréhension accrue des phénomènes liés à l'architecture du convertisseur.

Dans un second temps, nous avons étudié la faisabilité d'une implémentation à faibles ressources d'une extraction spectrale. En comparant les différentes techniques existantes et en proposant une nouvelle approche basée sur des opérations arithmétiques simples, nous sommes parvenus à un système consommant très peu de ressources logiques et capable d'extraire les premières harmoniques d'un signal converti. Cependant, le faible nombre d'harmoniques calculées par ces méthodes font qu'elles se destinent davantage à un contexte de type BIST. En effet, l'obtention d'une table de correction valable s'effectue à partir du calcul d'une trentaine de raies harmoniques.

Conclusion Générale

Rappel du contexte

La convergence des différents standards et des protocoles de communications numériques requiert la conception de convertisseurs analogique numérique de haute résolution et de fréquence d'échantillonnage très élevée. La conception de tels convertisseurs est pour le moment inaccessible à cause des sensibilités des architectures aux variations de procédés de fabrication. En attendant la maturité de la prochaine génération de CAN, les méthodes de compensation permettent à la fois d'améliorer la connaissance des points de blocage de la conception et de repousser les limites des convertisseurs actuels. Cette nouvelle compréhension fait partie du processus d'évolution des architectures et est donc primordiale à l'élaboration de procédés de conversions robustes.

Bilan des chapitres

Après avoir exposé les principes de base de la quantification analogique numérique, le premier chapitre a présenté différentes architectures afin de sensibiliser le lecteur aux difficultés de la conception de tels circuits. La mise en place de composants à la fois linéaires et rapides face aux vitesses de fonctionnement demandées est un challenge loin d'être résolu. Le compromis linéarité/vitesse impose de limiter les vitesses de fonctionnement au regard de la qualité de la conversion. L'idée de relaxer les contraintes de linéarité afin d'augmenter cette vitesse, requiert la compensation ces nouvelles non-linéarités.

Le second chapitre nous a permis de parcourir de façon globale mais non exhaustive, la plupart des méthodes dites de post-correction. Les méthodes les plus simples consistent en un échange de valeurs de conversion à travers une table de correspondance d'une à plusieurs dimensions. D'autres méthodes, plus mathématiques, tentent de modéliser le convertisseur à partir de modèles non-linéaires à mémoire. La complexité des calculs rend la mise ne oeuvre de ces méthodes particulièrement difficile pour un résultat très proche de celui obtenu avec les technique utilisant les tables de correspondance. Le dernier ensemble de compensation présenté dans le chapitre 2 est de loin le plus utilisé par les concepteurs de convertisseur. Lors de la conception, ces méthodes de compensation, telles que le dithering ou la calibration des capacités, sont implémentées afin de minimiser les effets des variations et dérives de procédés de fabrication. La meilleure solution consiste évidemment en un mélange subtil des différentes méthodes. Nous avons sélectionné pour notre étude, une méthode de compensation basée sur une table de correspondance uni-dimensionnelle, qui possède la propriété d'être facilement mise en oeuvre.

5.4. CONCLUSION

Le troisième chapitre expose différents moyens de calculer les valeurs des non-linéarités utilisées pour remplir la table de correspondance correctrice. Deux grandes méthodes s'offrent alors : la méthode statistique, et la méthode analytique. La première consiste à estimer la fréquence d'apparition de chaque code de sortie pour un signal d'entrée et de comparer ce résultat avec la réponse attendu. Cette méthode classique est actuellement utilisée pour le test et la caractérisation des convertisseurs. Elle permet une estimation assez précise des NLI mais requiert un très grand nombre d'échantillons. La méthode analytique se base sur une modélisation de la courbe représentant les non-linéarités. Les paramètres de ce modèle peuvent alors être calculés directement à partir du spectre complexe du signal converti. Cette méthode permet de réduire drastiquement le nombre d'échantillons nécessaires à l'estimation des NLI.

Dans le cadre de la compensation, deux possibilités s'offrent à nous : soit la compensation est effectuée de façon externe au composant, soit la compensation est intégrée au composant. Dans la premier cas, l'utilisateur dispose de toutes les ressources et du temps nécessaire. Le calcul des non-linéarités peut alors s'effectuer à partir de la méthode statistique. Par contre, dans le cadre d'une compensation embarquée, les ressources nécessaires en termes de taille de silicium ajouté, de complexité de calcul se doivent d'être minimales. Dans ce second cas, l'utilisation de méthodes d'extraction analytique semble plus adaptée.

Dans le quatrième chapitre, nous étudions de façon expérimentale la robustesse de la méthode de compensation. Diverses expériences sont menées afin de déterminer les paramètres influents sur la qualité de la compensation. Les conclusions de ces expériences valident l'utilisation d'une table de correction calculée pour une fréquence d'entrée et une fréquence d'échantillonnage fixes. La compensation n'est alors pas optimale pour toutes les gammes de fréquences utilisées, mais l'apport de la compensation est toujours significatif. L'idéal serait bien évidemment de recalculer la table de correspondance pour les différentes conditions d'utilisation, mais cela nécessiterait une reprogrammation du convertisseur au cours de sa vie. Cette pratique n'est donc pas réalisable mais l'utilisation d'une compensation embarquée permettrait de répondre à ce besoin.

Le cinquième chapitre aborde les prémices de la compensation embarquée. Après avoir exposé des simplifications liées à une nouvelle compréhension de la courbe des non-linéarités du convertisseur étudié, nous avons réduit les besoins de la méthode analytique d'extraction des non-linéarités. La seconde partie du chapitre étudie différentes solutions pour extraire les puissances des raies harmoniques du signal converti. Après avoir exposé le principe de fonctionnement de la transformée de Fourier rapide ainsi que de son implémentation séquentielle, deux techniques sont mises en avant : l'algorithme de Goertzel, et notre nouvel algorithme intitulé LoCHE. Ces deux techniques permettent de calculer à partir de très peu de ressources un nombre limité de raies harmoniques. Malheureusement, le faible nombre d'harmoniques calculables par ces deux méthodes les destinent à du test intégré qu'à de la compensation intégrée.

Perspectives

Au travers de ce manuscrit, de nombreuses techniques ont été étudiées, et mises en avant. Parmi elles, l'utilisation de l'analyse spectrale en vue de l'extraction des non-linéarités du convertisseur, nous a permis d'établir de nouvelles orientations d'études sur ces erreurs. La mise en place d'un formalisme fréquentiel associé aux courbes de NLI, nous semble être un sujet d'étude important à poursuivre à l'issue de

ces premiers travaux. La perspective majeure est de pouvoir relier ces phénomènes cycliques directement à l'architecture interne du convertisseur. Cette analyse pourrait permettre aux concepteurs de modifier leurs blocs analogiques afin de contrôler l'apparition des récurrences et donc de maîtriser les fluctuations de la courbe de NLI.

Les résultats de la compensation étudiée sont encourageants et nous incitent à poursuivre également dans cette voie. Pendant l'implémentation de cette technique, nous nous sommes aperçu que la quantification de la courbe de non-linéarités pouvait, elle aussi, être l'objet d'une étude plus poussée. Une quantification de type $\Sigma - \Delta$ pourrait permettre de modifier l'allure du Spectr_{inl} en repoussant sur les récurrences élevées, la majorité de l'énergie présente sur les récurrences faibles. Cette quantification pourrait s'effectuer avec suréchantillonnage et fournir deux tables permettant de diminuer l'erreur de quantification de $+/- 0.5\text{LSB}$ à $+/- 0.25\text{LSB}$. La correction devrait alors être améliorée. Quelques travaux ont été menés sur ce sujet, mais par faute de temps, il ne nous a pas été possible de les conduire plus en avant. Cependant, les premiers résultats semblaient confirmer l'amélioration attendue.

Conclusion finale

L'objectif de cette thèse était de réaliser une compensation des non-linéarités d'un convertisseur conçu pour pouvoir fonctionner à haute vitesse mais avec une linéarité dégradée. Après une étude des différentes méthodes de compensation, nous avons opté pour une compensation simple dont l'efficacité a été testée par la suite. La conclusion de cette étude est qu'il est possible d'améliorer la linéarité du composant en calculant une table de correspondances pour une fréquence d'entrée donnée. La compensation *offline* nous semble donc viable dans un contexte industriel. Il faudrait cependant étudier l'effet du vieillissement sur cette correction.

L'étude menée autour de la compensation *online* ne nous a pas permis de proposer une solution répondant aux objectifs. La difficulté d'extraire les non-linéarités du convertisseur pendant son fonctionnement est un problème majeur à cette réalisation. Cependant, notre recherche de réduction de la complexité a mis en avant l'étude fréquentielle des courbes de non-linéarités, et nous a également permis d'écrire un nouvel algorithme qui pourrait être particulièrement bien adapté à des architectures BIST basées sur l'analyse fréquentielle..

5.4. CONCLUSION

Bibliographie

- [AAG⁺02] F. Adamo, F. Attivissimo, N. Giaquinto, M. Savino, and D.E. ed Elettronica. FFT test of A/D converters to determine the integral nonlinearity. *Instrumentation and Measurement, IEEE Transactions on*, 51(5) :1050–1054, 2002.
- [ABB⁺01] F. Azais, S. Bernard, Y. Bertrand, X. Michel, and M. Renovell. A low-cost adaptive ramp generator for analog BIST applications. *VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001*, pages 266–271, 2001.
- [ADR04] P. Arpaia, P. Daponte, and S. Rapuano. A state of the art on ADC modelling. *Computer Standards Interfaces*, 26(1) :31–42, 2004.
- [AIW⁺87] Y. Akazawa, A. Iwata, T. Wakimoto, T. Kamato, H. Nakamura, and H. Ikawa. A 400MSPS 8b flash AD conversion LSI. *Solid-State Circuits Conference. Digest of Technical Papers. 1987 IEEE International*, 30, 1987.
- [AK75] A. Arbel and R. Kurz. Fast ADC. *IEEE Trans. Nucl. Sci*, 22(1) :446–451, 1975.
- [AKTS92] S. Abdennadher, S. Kiaei, G. Temes, and R. Schreier. Adaptive selfcalibrating delta-sigma modulators. *Electronics Letters*, 28(14) :1288–1289, 1992.
- [Alt05] Altera. Stratix II : Device Handbook. *Altera Corporation*, 2005.
- [BABR02] S. Bernard, F. Azais, Y. Bertrand, and M. Renovell. A high accuracy triangle-wave signal generator for on-chip ADC testing. *European Test Workshop, 2002. Proceedings. The Seventh IEEE*, pages 89–94, 2002.
- [BBGdE98] D. Bellan, A. Brandolini, A. Gandelli, and D. di Elettronica. ADC nonlinearities and harmonic distortion in FFT test. *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE*, 2, 1998.
- [BCA⁺30] S. Bernard, M. Comte, F. Azais, Y. Bertrand, and M. Renovell. A new methodology for adc test flow optimization. *Test Conference, 2003. Proceedings. ITC 2003. International*, 1, 1930.
- [BCA⁺04] S. Bernard, M. Comte, F. Azais, Y. Bertrand, and M. Renovell. Efficiency of Spectral-Based ADC Test Flows to Detect Static Errors. *Journal of Electronic Testing*, 20(3) :257–267, 2004.
- [BDR05] E. Balestrieri, P. Daponte, and S. Rapuano. A state of the art on ADC error compensation methods. *Instrumentation and Measurement, IEEE Transactions on*, 54(4) :1388–1394, 2005.

BIBLIOGRAPHIE

- [Ber04] DI Bergman. Dynamic error correction of a digitizer for time-domain metrology. *Instrumentation and Measurement, IEEE Transactions on*, 53(5) :1384–1390, 2004.
- [BIHR] N. Björnsell, M. Isaksson, P. Händel, and D. Rönnöw. Kautz-Volterra modelling of an analogue-to-digital converter using a stepped three-tone excitation.
- [Bla94] J. Blair. Histogram measurement of ADC nonlinearities using sine waves. *Instrumentation and Measurement, IEEE Transactions on*, 43(3) :373–383, 1994.
- [BSR86] M. Bossche, J. Schoukens, and J. Renneboog. Dynamic testing and diagnostics of A/D converters. *Circuits and Systems, IEEE Transactions on*, 33(8) :775–785, 1986.
- [BW88] BE Boser and BA Wooley. The design of sigma-delta modulation analog-to-digital converters. *Solid-State Circuits, IEEE Journal of*, 23(6) :1298–1308, 1988.
- [CA93] WT Colleran and AA Abidi. A 10-b, 75-MHz two-stage pipelined bipolar A/D converter. *Solid-State Circuits, IEEE Journal of*, 28(12) :1187–1199, 1993.
- [CG95] TB Cho and PR Gray. A 10 b, 20 Msample/s, 35 mW pipeline A/D converter. *Solid-State Circuits, IEEE Journal of*, 30(3) :166–172, 1995.
- [CJ] N. Csizmadia and A. Janssen. Estimating the Integral Non-Linearity of AD-Converters via the Frequency Domain. *log*, 954 :30.
- [CT65] J.W. Cooley and J.W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19(90) :297–301, 1965.
- [CT92] J.C. Candy and G.C. Temes. Oversampling methods for A/D and D/A conversion, 1992.
- [CT96] G. Cauwenberghs and GC Temes. Adaptive calibration of multiple quantization over-sampled A/Dconverters. *Circuits and Systems, 1996. ISCAS'96., 'Connecting the World'., 1996 IEEE International Symposium on*, 1, 1996.
- [CT00] G. Cauwenberghs and GC Temes. Adaptive digital correction of analog errors in MASH ADCs. I. Off-line and blind on-line calibration. *Circuits and Systems II : Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II : Express Briefs, IEEE Transactions on]*, 47(7) :621–628, 2000.
- [Dan40] G.C. Danielson. *Improvements in Practical Fourier Analysis and Their Application to X-Ray Scattering from Liquids*. PhD thesis, PURDUE UNIVERSITY, 1940.
- [FBD⁺06] V. Fresnaud, L. Bossuet, D. Dallet, S. Bernard, J.M. Janik, B. Agnus, P. Cauvet, and P. Gandy. A Low Cost Alternative Method for Harmonics Estimation in a BIST context. *Test Symposium, 2006. ETS'06. Eleventh IEEE European*, pages 193–198, 2006.
- [FNCS04] M.G.C. Flores, M. Negreiros, L. Carro, and AA Susin. INL and DNL estimation based on noise for ADC test. *Instrumentation and Measurement, IEEE Transactions on*, 53(5) :1391–1395, 2004.
- [Fou22] J.B.J. Fourier. *Théorie analytique de la chaleur*. Chez Firmin Didot, père et fils..., 1822.
- [FS98] MP Flynn and B. Sheahan. A 400-Msample/s, 6-b CMOS folding and interpolating ADC. *Solid-State Circuits, IEEE Journal of*, 33(12) :1932–1938, 1998.
- [Ger78] A. Gersho. Principles of quantization. *Circuits and Systems, IEEE Transactions on*, 25(7) :427–436, 1978.

BIBLIOGRAPHIE

- [Gil82] B. Gilbert. A monolithic microsystem for analog synthesis of trigonometric functions and their inverses. *Solid-State Circuits, IEEE Journal of*, 17(6) :1179–1191, 1982.
- [GKHK91] Y. Gendai, Y. Komatsu, S. Hirase, and M. Kawata. An 8b 500MHz ADC. *Solid-State Circuits Conference, 1991. Digest of Technical Papers. 38th ISSCC., 1991 IEEE International*, pages 172–311, 1991.
- [Goe58] G. Goertzel. An Algorithm for the Evaluation of Finite Trigonometric Series. *The American Mathematical Monthly*, 65(1) :34–35, 1958.
- [Gra90] RM Gray. Quantization noise spectra. *Information Theory, IEEE Transactions on*, 36(6) :1220–1244, 1990.
- [HIUK86] T. Hayashi, Y. Inabe, K. Uchimura, and T. Kimura. A multistage delta-sigma modulator without double integration loop. *Solid-State Circuits Conference. Digest of Technical Papers. 1986 IEEE International*, 29, 1986.
- [HJ85] C. Hogge Jr. A self correcting clock recovery circuit. *Lightwave Technology, Journal of*, 3(6) :1312–1314, 1985.
- [HSP00] P. Handel, M. Skoglund, and M. Pettersson. A calibration scheme for imperfect quantizers. *Instrumentation and Measurement, IEEE Transactions on*, 49(5) :1063–1068, 2000.
- [HZHvR06] P. Harpe, A. Zanicopoulos, H. Hegt, and A. van Roermund. Digital post-correction of front-end track-and-hold circuits in ADCs. *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, page 4, 2006.
- [IHK91] FH Irons, DM Hummels, and SP Kennedy. Improved compensation for analog-to-digital converters. *Circuits and Systems, IEEE Transactions on*, 38(8) :958–961, 1991.
- [ISM⁺84] M. Inoue, H. Sadamatsu, A. Matsuzawa, A. Kanda, and T. Takemoto. A monolithic 8-bit A/D converter with 120 MHz conversion rate. *Solid-State Circuits, IEEE Journal of*, 19(6) :837–841, 1984.
- [JF07] J.M. Janik and V. Fresnaud. A spectral approach to estimate the INL of A/D converter. *Computer Standards Interfaces*, 29(1) :31–37, 2007.
- [KBCJ06] V. Kerzérho, S. Bernard, P. Cauvet, and J.M. Janik. A First Step for an INL Spectral-Based BIST : The Memory Optimization. *Journal of Electronic Testing*, 22(4) :351–357, 2006.
- [KBJC05] V. Kerzérho, S. Bernard, J.M. Janik, and P. Cauvet. Comparison Between Spectral-Based Methods for INL Estimation and Feasibility of Their Implantation. *IEEE International Mixed-Signal Testing Workshop*, pages 270–275, 2005.
- [KMNS93] H. Kimura, A. Matsuzawa, T. Nakamura, and S. Sawada. A 10-b 300-MHz interpolated-parallel A/D converter. *Solid-State Circuits, IEEE Journal of*, 28(4) :438–446, 1993.
- [KSW⁺00] P. Kiss, J. Silva, A. Wiesbauer, T. Sun, U.K. Moon, J.T. Stonick, and G.C. Temes. Adaptive digital correction of analog errors in MASH ADC's- Part II : Correction using test-signal injection. *IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing*, 47(7) :629–638, 2000.

BIBLIOGRAPHIE

- [KY79] S. Klein and S. Yasui. Nonlinear systems analysis with non-Gaussian white stimuli ; General basis functionals and kernels (Corresp.). *Information Theory, IEEE Transactions on*, 25(4) :495–500, 1979.
- [LASH] H. Lundin, T. Andersson, M. Skoglund, and P. Handel. Analog-to-digital converter error correction using frequency selective tables. *RadioVetenskap och Kommunikation (RVK)*, pages 487–490.
- [LS65] YW Lee and M. Schetzen. Measurement of the Wiener Kernels of a Non-linear System by Cross-correlation. *International Journal of Control*, 2(3) :237–254, 1965.
- [LS90] TC Leslie and B. Singh. An improved sigma-delta modulator architecture. *Circuits and Systems, 1990., IEEE International Symposium on*, pages 372–375, 1990.
- [LSH04] H. Lundin, M. Skoglund, and P. Handel. A criterion for optimizing bit-reduced post-correction of AD converters. *Instrumentation and Measurement, IEEE Transactions on*, 53(4) :1159–1166, 2004.
- [LSH05] H. Lundin, M. Skoglund, and P. Handel. Optimal index-bit allocation for dynamic post-correction of analog-to-digital converters. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 53(2) :660–671, 2005.
- [LSMW⁺04] Y. Li, J.E. Silva-Martinez, D.M. Walker, K. Chang, and E. Sanchez-Sinencio. Design of high speed folding and interpolating analog-to-digital converter. 2004.
- [LSS00] Y. Li and E. Sanchez-Sinencio. Current mirror based folding amplifier. *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on*, 1, 2000.
- [Lun03] H. Lundin. Post-correction of analog-to-digital converters. *Royal Institute of Technology (KTH), Stockholm, Sweden, Licentiate thesis TRITA–S3–SB–0324, May*, 2003.
- [Mah87] M. Mahoney. *DSP-based Testing of Analog and Mixed-signal Circuits : Tutorial*. IEEE Computer Society, 1987.
- [MCD⁺00] C. Morandi, G. Chiorboli, D. Dallet, D. Haddadi, S. Mazzoleni, J. Machado da Silva, H. Pernull, and PY Roy. DYNAD : a Framework IV SMT project addressed to the development of dynamic test techniques for analog-to-digital converters. *Computer Standards and Interfaces*, 22(2) :113–119, 2000.
- [MotIIMS01] Waveform Measurement, Analysis Technical Committee of the IEEE Instrumentation, and USA Measurement Society. IEEE 1241-2000 standard for terminology and test methods for analog-to-digital converters. *IEEE Standard*, 55(1), 2001.
- [Mou89] D. Moulin. Real-time equalization of A/D converter nonlinearities. *Circuits and Systems, 1989., IEEE International Symposium on*, pages 262–267, 1989.
- [MŠ] P. Mikulik and J. Šaliga. Volterra Filtering for ADC Error Correction.
- [MS02] P. Mikulik and J. Saliga. Volterra filtering for integrating ADC error correction, based on an a priori error model. *Instrumentation and Measurement, IEEE Transactions on*, 51(4) :870–875, 2002.

BIBLIOGRAPHIE

- [NNI⁺04] H. Nosaka, M. Nakamura, M. Ida, K. Kurishima, T. Shibata, M. Tokumitsu, and M. Muraguchi. A 24-Gsps 3-bit Nyquist ADC using InP HBTs for electronic dispersion compensation. *Microwave Symposium Digest, 2004 IEEE MTT-S International*, 1, 2004.
- [P⁺92] W.H. Press et al. *Numerical recipes in C*. Cambridge University Press Cambridge, 1992.
- [Pac] H. Packard. Dynamic Performance Testing of A to D Converters. *HP Product Note No5180A*, 2.
- [PDW89] MJM Pelgrom, ACJ Duinmaijer, and APG Welbers. Matching properties of MOS transistors. *Solid-State Circuits, IEEE Journal of*, 24(5) :1433–1439, 1989.
- [PG79] RJ Plassche and RJ Grift. A high-speed 7-bA/D converter. *IEEE Journal of Solid-State Circuits*, 14 :938–943, 1979.
- [RDM04] C. Rebai, D. Dallet, and P. Marchegay. Signal generation using single-bit sigma-delta techniques. *Instrumentation and Measurement, IEEE Transactions on*, 53(4) :1240–1244, 2004.
- [RI87] TA Rebold and FH Irons. A phase-plane approach to the compensation of high-speed analog-to-digital converters. *IEEE International Symposium on Circuits and Systems, Philadelphia, PA, May*, 1987.
- [RS96] R. Roovers and MSJ Steyaert. A 175 Ms/s, 6 b, 160 mW, 3.3 V CMOS A/D converter. *Solid-State Circuits, IEEE Journal of*, 31(7) :938–944, 1996.
- [SAW89] M. Shinagawa, Y. Akazawa, and T. Wakimoto. Jitter analysis of high speed sampling systems. *VLSI Circuits, 1989. Digest of Technical Papers., 1989 Symposium on*, pages 95–96, 1989.
- [Sch64] L. Schuchman. Dither Signals and Their Effect on Quantization Noise. *IEEE Transactions on Communications*, 12(4) :162–165, 1964.
- [Sch74] M. Schetzen. A theory of non-linear system identification. *International Journal of Control*, 20(4) :577–592, 1974.
- [Sch81] M. Schetzen. Nonlinear system modeling based on the Wiener theory. *Proceedings of the IEEE*, 69(12) :1557–1573, 1981.
- [SG88] S. Sutarja and PR Gray. A pipelined 13-bit 250-ks/s 5-V analog-to-digital converter. *Solid-State Circuits, IEEE Journal of*, 23(6) :1316–1323, 1988.
- [SHT⁺00] L. Singer, S. Ho, M. Timko, D. Kelly, A.D. Inc, and MA Wilmington. A 12 b 65 MSample/s CMOS ADC with 82 dB SFDR at 120 MHz. *Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International*, pages 38–39, 2000.
- [SN97] S.K. Sunter and N. Nagi. A Simplified Polynomial-Fitting Algorithm for DAC and ADC BIST. *Proc. International Test Conference*, pages 389–395, 1997.
- [TL96] J. Tsimbinos and KV Lever. Error Table And Volterra Compensation Of A/d Converter Nonlinearities-A Comparison. *Signal Processing and Its Applications, 1996. ISSPA 96., Fourth International Symposium on*, 2, 1996.

BIBLIOGRAPHIE

- [TLS⁺93] J. Tsimbinos, KV Lever, D. Sci, T. Organisation, and V. Melbourne. Applications of higher-order statistics to modelling, identification and cancellation of nonlinear distortion in high-speedsamplers and analogue-to-digital converters using the Volterra andWiener models. *Higher-Order Statistics, 1993. IEEE Signal Processing Workshop on*, pages 379–383, 1993.
- [TLS⁺96] J. Tsimbinos, KV Lever, D. Sci, T. Organ, and SA Salisbury. Computational complexity of Volterra based nonlinear compensators. *Electronics Letters*, 32(9) :852–854, 1996.
- [TLS⁺97] J. Tsimbinos, KV Lever, D. Sci, T. Organ, and SA Salisbury. Improved error-table compensation of A/D converters. *Circuits, Devices and Systems, IEE Proceedings-*, 144(6) :343–349, 1997.
- [TMBSL02] J. Tsimbinos, W. Marwood, A. Beaumont-Smith, and CC Lim. Results of A/D converter compensation with a VLSI chip. *Information, Decision and Control, 2002. Final Program and Abstracts*, pages 289–293, 2002.
- [TSE98] S. Tsukamoto, WG Schofield, and T. Endo. A CMOS 6-b, 400-MSample/s ADC with error correction. *Solid-State Circuits, IEEE Journal of*, 33(12) :1939–1947, 1998.
- [TSH06] M. Taherzadeh-Sani and AA Hamoui. Digital background calibration of interstage-gain and capacitor-mismatch errors in pipelined ADCs. *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, page 4, 2006.
- [Tsi95] J. Tsimbinos. *Identification and compensation of nonlinear distorsion*. PhD thesis, University of South Australia, 1995.
- [vdP03] R.J. van de Plassche. *CMOS Integrated Analog-To-Digital and Digital-To-Analog Converters*. Kluwer Academic Publishers, 2003.
- [vdPB88] RJ van de Plassche and P. Baltus. An 8-bit 100-MHz full-Nyquist analog-to-digital converter. *Solid-State Circuits, IEEE Journal of*, 23(6) :1334–1344, 1988.
- [Vol87] V. Volterra. Sopra le funzioni che dipendono da altre funzioni, *Atti Reale Accad. Lincei Rend.(4)*, 3(2) :97–105, 1887.
- [VR97] P. Vorenkamp and R. Roovers. A 12-b, 60-MSample/s cascaded folding and interpolating ADC. *Solid-State Circuits, IEEE Journal of*, 32(12) :1876–1886, 1997.
- [VvdP96] A.G.W. Venes and R.J. van de Plassche. An 80-MHz, 80-mW, 8-b CMOS Folding AID Converter with Distributed Track-and-Hold Preprocessing. *IEEE Journal of Solid-states Circuit*, 31(12), 1996.
- [Xu99] F. Xu. A New Approach for the Nonlinearity Test of ADCs/DACs and its Application for BIST. *IEEE European Test Workshop*, 99 :34–38, 1999.
- [YANY84] Y. Yoshii, K. Asano, M. Nakamura, and C. Yamada. An 8b 100MS/s flash ADC. *Solid-State Circuits Conference. Digest of Technical Papers. 1984 IEEE International*, 27, 1984.

Annexe A

Démonstration mathématique de la méthode par Histogramme

Le calcul des non-linéarités du convertisseur à partir de l'histogramme nécessite la connaissance au préalable d'un histogramme de référence. Celui-ci est obtenu mathématiquement à partir d'une conversion idéale. La densité de probabilité du signal d'entrée est tout d'abord nécessaire. Dans le cas d'un signal sinusoïdal $V(t)$, de pulsation ω et d'amplitude A , la densité de probabilité $f(V(t))$ est donnée par la formule suivante :

$$f(V(t)) = \frac{1}{\pi\sqrt{A^2 - (V(t))^2}} \quad (5.44)$$

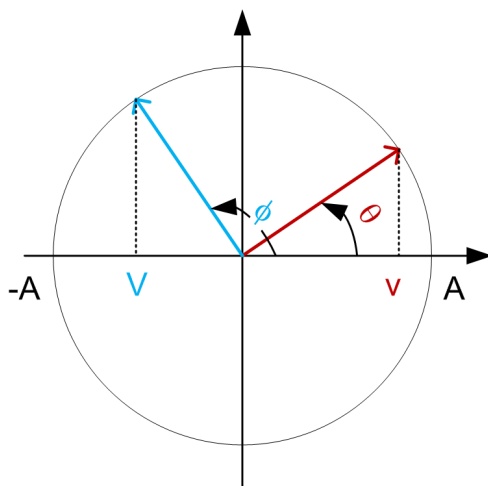


FIG. 5.24 – Représentation du signal sinusoïdal

Sa justification provient de la démonstration suivante. Il faut tout d'abord considérer le signal sinusoïdal sous sa forme trigonométrique représentée sur le cercle de la figure 5.24. La probabilité d'avoir une tension V inférieure à la valeur de référence v est donnée par le rapport de l'intervalle des angles favorables à l'obtention de ces valeurs sur le nombre total d'angles possibles. Dans notre cas, pour que le signal soit inférieur à v , il faut que l'angle ϕ soit compris entre θ et $2\pi - \theta$. La probabilité $P(V < v)$ est donc donnée par la probabilité que ϕ soit dans cet intervalle. Comme ϕ dépend du temps, et que cette variable

BIBLIOGRAPHIE

est linéaire, la probabilité est donc tout simplement : $P(V < v) = \frac{\pi - \theta}{\pi}$. De plus comme $\theta = \arcsin\left(\frac{v}{A}\right)$, on obtient la formule suivante :

$$P(V < v) = \frac{\pi - \arcsin\left(\frac{v}{A}\right)}{\pi} \quad (5.45)$$

Or, comme la densité de probabilité d'une variable aléatoire est relié à sa probabilité par la formule suivante :

$$f(v) = \frac{dP(V < v)}{dv} \quad (5.46)$$

on obtient bien l'équation précédemment énoncée.

$$f(v) = \frac{1}{\pi\sqrt{A^2 - v^2}} \quad (5.47)$$

A partir de cette représentation mathématique, on peut désormais calculer la probabilité théorique $p_{th}(i)$ d'apparition du code i . Celle-ci est donnée par l'aire présente sous la courbe de densité de probabilité et comprise entre les tensions de transition V_i et V_{i+1} (figure 5.25).

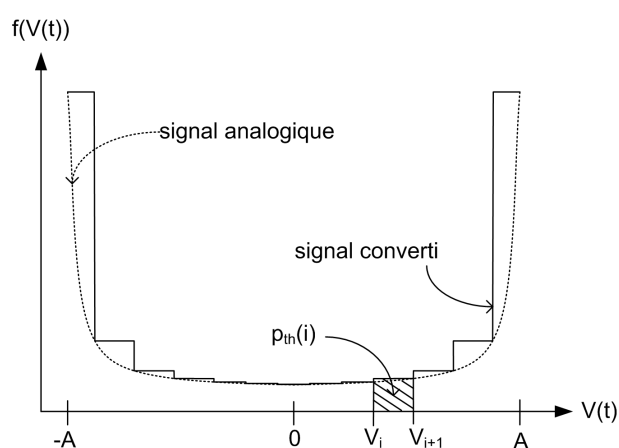


FIG. 5.25 – Représentation du signal sinusoïdal

Le calcul de cette probabilité s'exprime sous la forme suivante :

$$p_{th}(i) = \int_{V_i}^{V_{i+1}} f(V(t)) dV(t) \quad (5.48)$$

C'est cette probabilité qu'il convient de comparer à la mesure effectuée par histogramme sur le signal converti réel. En effet, le calcul de l'histogramme revient à compter les occurrences d'apparition des codes à l'issue de la conversion. Ce nombre d'occurrences est donc en relation directe avec leur probabilité d'apparition, ce qui se traduit par l'équation suivante :

$$p(i) = \frac{H(i)}{S} \quad (5.49)$$

où $H(i)$ représente le nombre d'occurrences du code i dans l'histogramme mesuré H , et S étant le nombre d'échantillons utilisés.

L'égalité de $p_{th}(i)$ et $p(i)$ est réalisée quand le convertisseur ne possède pas de non-linéarité. A partir de la mesure $p(i)$ et de l'expression de $p_{th}(i)$, les valeurs des tensions de transition peuvent être retrouvées. Pour simplifier les calculs, l'utilisation de l'histogramme cumulé est préféré. Cet histogramme, H_c , est construit directement à partir de l'histogramme H de la façon suivante :

$$H_c(i) = \sum_{j \leq i} H(j) \quad (5.50)$$

D'un point de vue probabilité, $H_c(i)$ représente alors la somme des occurrences inférieures ou égales au code i . On note cette nouvelle probabilité $p_c(i)$, elle s'exprime avec la formule suivante :

$$p_c(i) = \int_{-A}^{V_{i+1}} f(V(t)) dV(t) \quad (5.51)$$

et se calcule à partir de l'histogramme H selon :

$$p_c(i) = \sum_{j \leq i} \frac{H(j)}{S} \quad (5.52)$$

Ces nouvelles équations permettent d'extraire la valeur de la tension de transition :

$$V_{i+1} = -A \cdot \cos(\pi p_c(i)) \quad (5.53)$$

Dans la pratique, ces valeurs sont souvent normalisées (V_i^{norm}) afin d'éliminer les problèmes liés à l'estimation de l'amplitude A et de l'offset éventuel du signal d'entrée. Cette normalisation impose alors le calcul d'un quantum de référence q_r dont la valeur est calculée à partir de la moyenne des $2^n - 3$ quantums situés entre les tensions de transition V_1^{norm} et $V_{2^n-2}^{norm}$. Les codes extrêmes sont volontairement écartés du fait de la saturation réalisée à l'entrée, afin d'être certain d'exciter tout les codes. Le calcul des non-linéarités s'effectue alors directement avec les formules suivantes :

$$NLD(i) = \frac{V_{i+1}^{norm} - V_i^{norm}}{q_r} - 1 \quad (5.54)$$

$$NLI(i) = \frac{V_i^{norm} - V_0^{norm}}{q_r} - i \quad (5.55)$$

Annexe B

Présentation de l'algorithme de FFT

Danielson et Lanczos démontrèrent que toute transformation de Fourier discrète $F_N(\cdot)$ de longueur N , pouvait être réécrite comme la somme de deux transformations de Fourier discrètes de longueur $\frac{N}{2}$: la première, $F_{N/2}^e(k)$, contenant les échantillons numérotés pairs, et la seconde, $F_{N/2}^o(k)$, contenant les échantillons numérotés impairs. La démonstration est effectuée ci-dessous :

$$\begin{aligned}
 F_N(k) &= \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2\pi k j}{N}} \\
 &= \sum_{j=0}^{N/2-1} f_{2j} \cdot e^{\frac{2\pi k 2j}{N}} + \sum_{j=0}^{N/2-1} f_{2j+1} \cdot e^{\frac{2\pi k (2j+1)}{N}} \\
 &= \sum_{j=0}^{N/2-1} f_{2j} \cdot e^{\frac{2\pi k j}{N/2}} + e^{\frac{2\pi k}{N}} \cdot \sum_{j=0}^{N/2-1} f_{2j+1} \cdot e^{\frac{2\pi k j}{N/2}} \\
 &= \sum_{j=0}^{N/2-1} f_{2j} \cdot W_{N/2}^{kj} + W_N^k \cdot \sum_{j=0}^{N/2-1} f_{2j+1} \cdot W_{N/2}^{kj} \\
 &= F_{N/2}^e(k) + W_N^k \cdot F_{N/2}^o(k)
 \end{aligned} \tag{5.56}$$

Il faut noter que le nombre complexe W possède des propriétés de symétrie qui permettent de simplifier les écritures. La première d'entre elles, est la suivante :

$$\begin{aligned}
 W_{N/2}^{k+N/2} &= W_{N/2}^k \cdot W_{N/2}^{N/2} \\
 &= W_{N/2}^k
 \end{aligned} \tag{5.57}$$

Il n'y a donc pas de contradiction de calculer $F_{N/2}^e(k)$ et $F_{N/2}^o(k)$ pour $k \in [0..N[$. Graphiquement, la décomposition de 2^3 échantillons de données se traduit par la figure 5.26. La décomposition peut être poursuivie sur les termes $F_{N/2}^e(k)$ et $F_{N/2}^o(k)$. Quatre nouvelles transformées de Fourier discrètes de longueur $\frac{N}{4}$ apparaissent : ce sont les décompositions paires et impaires des premières décompositions.

$$F_{N/2}^e(k) = F_{N/4}^{ee}(k) + W_{N/2}^k \cdot F_{N/4}^{eo}(k) \tag{5.58}$$

et

$$F_{N/2}^o(k) = F_{N/4}^{oe}(k) + W_{N/2}^k \cdot F_{N/4}^{oo}(k) \tag{5.59}$$

Une nouvelle propriété remarquable du nombre complexe W est introduite :

$$W_{N/2}^k = W_N^{2k} \tag{5.60}$$

Cette propriété permet d'exprimer les nouvelles décompositions en fonction de W_N et non plus $W_{N/2}$:

$$F_{N/2}^e(k) = F_{N/4}^{ee}(k) + W_N^{2k} \cdot F_{N/4}^{eo}(k) \tag{5.61}$$

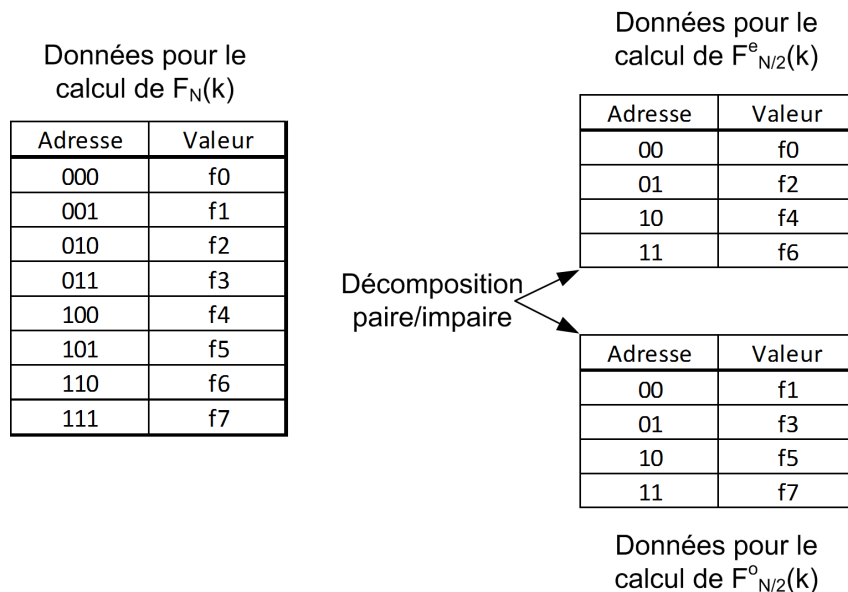


FIG. 5.26 – Décomposition de $F_N(k)$ en $F_{N/2}^e(k)$ et $F_{N/2}^o(k)$

et

$$F_{N/2}^o(k) = F_{N/4}^{oe}(k) + W_N^{2k} \cdot F_{N/4}^{oo}(k) \tag{5.62}$$

La décomposition peut alors être encore une fois appliquée dans notre exemple à 2^3 échantillons. Les

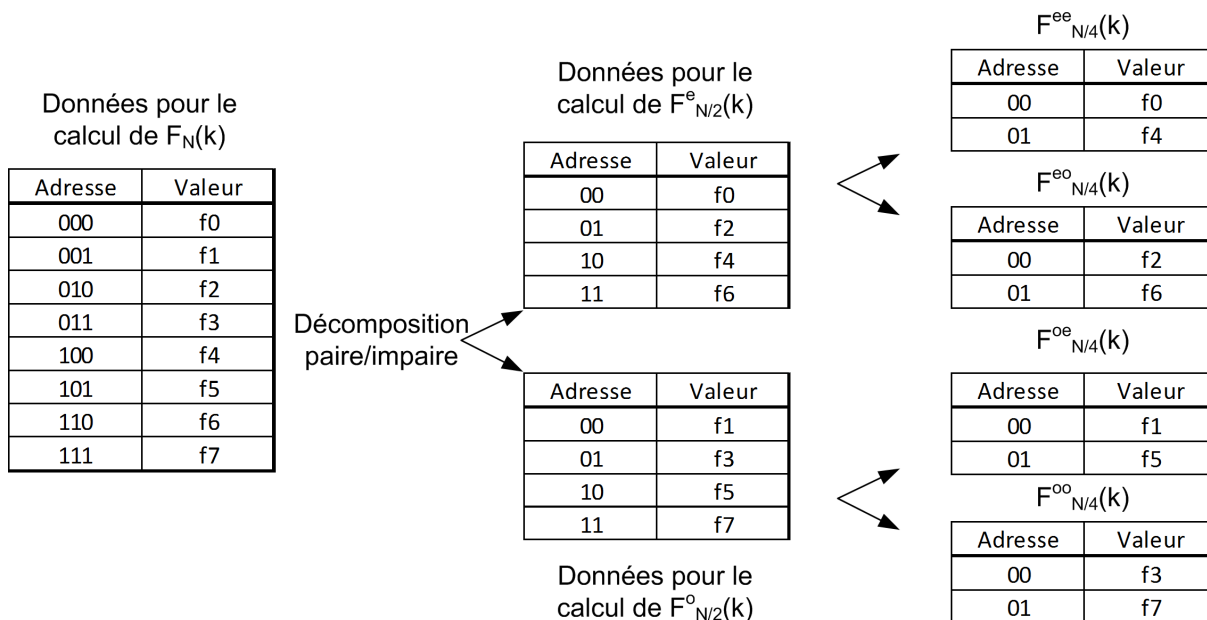


FIG. 5.27 – Seconde décomposition

nouvelles transformées de Fourier discrète sont alors dégénérées au simple échantillon. Par exemple, la

décomposition de $F_{N/4}^{ee}(k)$ fournit :

$$F_{N/4}^{ee}(k) = f_0 + W_N^{4k} \cdot f_4 \tag{5.63}$$

$F_N(k)$ peut donc être calculé de façon récursive, en fonction des nombres complexes W_N et des échantillons de départ. Pour cela, une première réorganisation des données doit être effectuée. Pour notre exemple à 2^3 échantillons, on observe que la réorganisation des données correspond à une opération dite de *bit reversing* effectuée sur l'adresse de l'échantillon. La figure 5.28 illustre ce propos. Un nouvel

Données réordonnées pour le calcul de $F_N(k)$

Adresse finale	Valeur	Adresse initiale
000	f0	000
001	f4	100
010	f2	010
011	f6	110
100	f1	001
101	f5	101
110	f3	011
111	f7	111

FIG. 5.28 – Réorganisation des échantillons

opérateur complexe est introduit : l'opérateur papillon. Cet opérateur est représenté sur la figure 5.29. A partir de deux valeurs d'entrées complexes E_0 et E_1 , et d'un paramètre complexe W_N^k , l'opérateur fournit les valeurs de sorties complexes S_0 et S_1 . Le calcul de ces deux valeurs est fourni par les équations suivantes :

$$\begin{aligned} S_0 &= E_0 + W_N^k \cdot E_1 \\ S_1 &= E_0 - W_N^k \cdot E_1 \end{aligned} \tag{5.64}$$

En introduisant l'opérateur papillon dans le calcul de la transformée de Fourier, on peut observer le

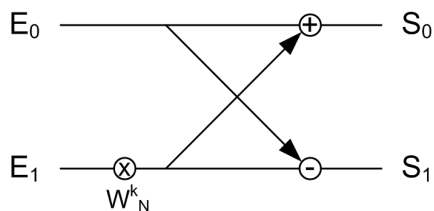


FIG. 5.29 – Opérateur papillon

parallélisme qui représente la force du calcul de l'algorithme. La figure 5.30 illustre le calcul sur 2^3 échantillons. On observe trois étages d'opérateurs papillons qui sont appelés séquentiellement. Pour 2^n échantillons, l'algorithme contiendrait $\log_2(2^n) = n$ étages constitués de 2^n multiplications.

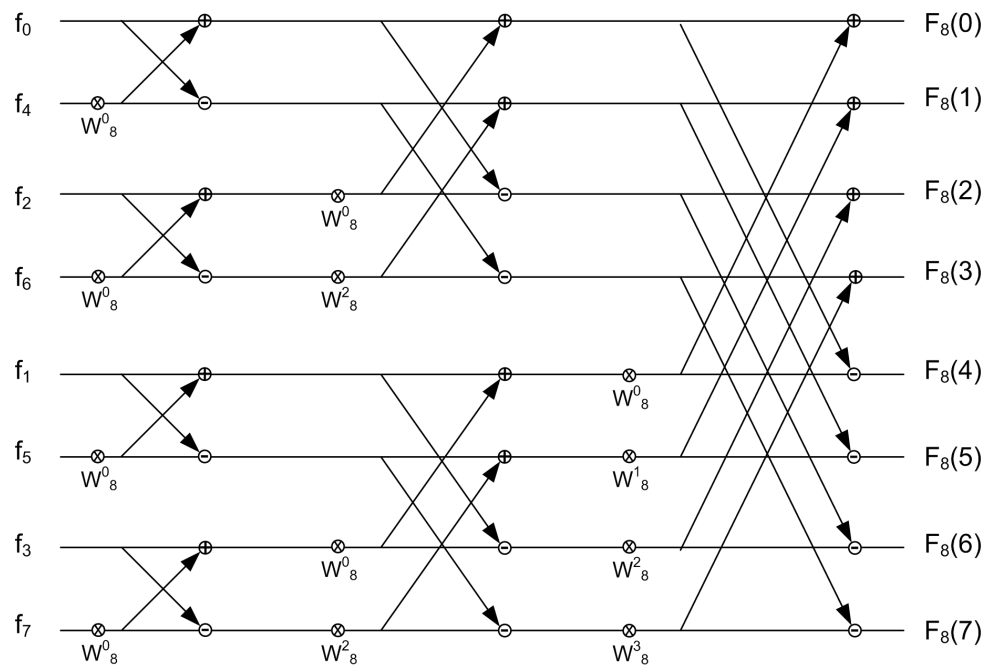


FIG. 5.30 – Calcul récursif et parallèle

Resumé

De nombreuses recherches tentent d'améliorer les convertisseurs actuels en proposant de nouvelles architectures et de nouveaux procédés de fabrication. Cette évolution est longue et doit s'exécuter étape par étape. Cependant, il est possible d'utiliser des méthodes permettant de compenser les lacunes d'un composant donné avant de franchir l'étape suivante. Ces méthodes de compensation permettent de repousser les limites du composant en attendant la maturité de la génération suivante. Elles permettent également de mieux comprendre les défauts actuels et d'orienter les concepteurs vers des pistes prometteuses pour leurs recherches.

C'est dans ce contexte que nous proposons d'étudier l'effet d'une compensation par table de correspondance uni-dimensionnelle (LUT 1D) sur un convertisseur de type à repliement et à interpolation.

Afin de remplir cette table de correction, nous proposons d'utiliser et d'optimiser un algorithme d'extraction des non-linéarités du composant, basé sur une analyse fréquentielle du signal converti.

Les paramètres sensibles de la méthode de compensation sont ensuite étudiées au travers d'expérimentations menées sur un convertisseur spécialement conçu pour cet objectif. Nous établissons qu'il est possible de calculer une table de correspondance suffisamment robuste indépendamment des variations de fréquences et de température.

Enfin, nous proposons une nouvelle méthode d'extraction des paramètres spectraux d'un signal à partir de ressources de calculs très faibles. Cette étude permet d'entamer le processus d'embarquement de la compensation au sein du convertisseur. Cette finalité fait partie des perspectives liées à cette thèse.

Mots clés : CAN, compensation, FFT, LoCHE, Goertzel, LUT, FPGA

Abstract

Many researches are trying to improve the nowadays analog-to-digital converters in proposing new architectures and new fab processes. This evolution is long and need to follow a step-by-step roadmap. However, one can use methods to compensate the lacks of a device while waiting for the next step. Those methods allow to push further the device's limits while waiting for the next generation. They also allow to have a better understanding of the defects that can drive the designers in new ways of designing.

In this context, we study the effect of a compensation based on a unidimensional LookUp Table (LUT 1D) on a folding and interpolation converter.

In order to fill this LUT, we present an optimized algorithm that can extract the non linearities of the device from a complex spectrum lead on the output data.

Sensitives parameters are then studied through a set of experiments lead on a dedicated converter. We prove that a unique LUT could be computed to provide a robust compensation independantly of the frequencies and temperature variations.

At last, we propose a new algorithm that perfoms the spectral extraction of a signal with low cost resources. This study allows us to begin the path of the embedded compensation.

Keywords : ADC, compensation, FFT, LoCHE, Goertzel, LUT, FPGA