



Joaquim
Estopinan
SICOM



2018/2019
13/05/19 – 02/08/19

Rapport de Stage 2A

Algorithme de test estimant la linéarité d'un
convertisseur analogique numérique cyclique

Tuteur: Matthieu Dubois
Matthieu.Dubois@pyxalis.com
+33 4 26 78 48 41

Pyxalis 2019

170 Rue de Chatagnon, 38430 Moirans

TABLE DES MATIERES

I. Introduction	3
1. Présentation de Pyxalis	3
2. Contexte	3
3. Objectif du stage	4
4. Linéarité statique d'un ADC.....	6
5. Fonctionnement d'un ADC cyclique	7
6. Description du modèle	8
II. Reconstruction analogique	9
1. Principe.....	9
2. Mise en place de la méthode	10
3. Prise en compte du bruit	11
4. Résultats	12
5. Architecture du code	15
III. Reconstruction numérique	16
1. Principe.....	16
2. Architecture du code	17
3. Retour sur le Mapping.....	18
4. Sélection des transitions	20
5. Création du signal d'entrée	21
6. Mesure des largeurs de codes.....	21
7. Résultats	22
IV. Conclusion	24
V. Références	25
VI. Annexes	26
1.....	26
VII. Résumés	27
1. Français.....	27
2. Anglais	27

I. Introduction

1. Présentation de Pyxalis

Pyxalis est une entreprise de taille humaine dont le cœur de métier est le développement de capteurs d'images. Avec une trentaine d'employés et une dynamique importante, les bureaux situés à Moirans constituent un lieu de travail agréable. L'open-space est divisé en secteurs d'activité : management, administration, achat & vente, conception analogique, conception numérique, layout et validation, dessin de pixels, caractérisation, prospection et support client.

Ce dernier pôle, fondamental pour l'entreprise fournit des solutions adaptées aux besoins de ses clients, quelque que soit leur taille. En effet, comme me l'a précisé Bertrand - Ingénieur Application – les grosses entreprises du secteur ne viseront que les clients importants alors que la force de Pyxalis est de travailler à l'écoute d'acteurs parfois plus modestes mais prometteurs.

L'entreprise propose des capteurs « *full-custom* », i.e. développés uniquement pour une application précise d'un client, et d'autres « *semi-custom* » qui pourront être utilisés pour les différents marchés cibles de Pyxalis : l'aérospatiale, le médical ou encore l'industrie automobile.

2. Contexte

La fonction d'un capteur d'image numérique est de constituer une représentation numérique fidèle d'une scène réelle. Pour se faire, la lumière de la scène visée est capturée puis transformée selon les étapes de la *figure 1* ci-dessous :

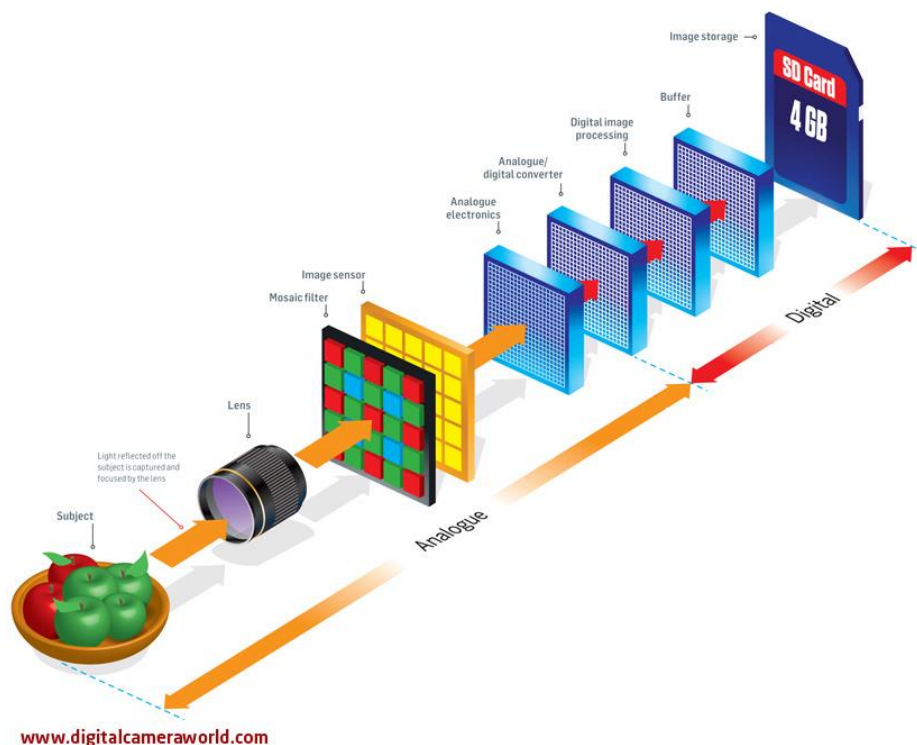


Figure 1 : Comment transformer la lumière en un fichier digital ? [4]

Pour commencer, la lumière incidente est focalisée sur le capteur d'image à l'aide de l'objectif. Un filtre couleur mosaïque est placé avant le capteur car ce dernier n'est sensible qu'à l'intensité lumineuse qui lui parvient, et non à la couleur. Ainsi, l'intensité lumineuse est mesurée pour chaque couleur primaire indépendamment. En sortie du capteur d'image et de l'électronique analogique qui l'entoure, il y a des tensions proportionnelles aux intensités lumineuses mesurées. Pour passer dans le monde numérique, il faut maintenant quantifier ces tensions avec des valeurs discrètes. C'est la tâche du Convertisseur Analogique Numérique (ADC). En sortie de l'ADC, un bloc de traitement numérique adapte le format des données délivrées par l'ADC et reconstitue les couleurs. L'image numérique est ensuite bufferisée, prête à être enregistrée sur un support externe.

Dans cette chaîne d'étapes pour numériser une image, le temps de développement de l'ADC est coûteux car la modélisation de l'interface entre l'analogique et le numérique requiert de longues simulations. La *figure 2* montre différentes architectures d'ADC avec des couples de performances résolution/vitesse différents. Le cas d'étude de ce stage porte sur un ADC dit cyclique (moyenne résolution, moyenne fréquence).

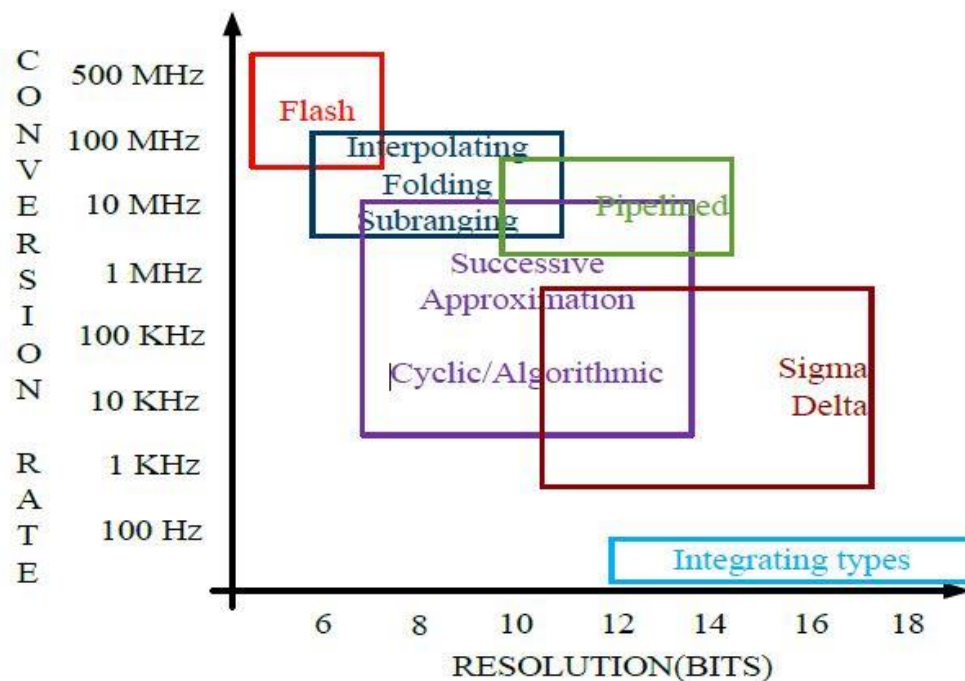


Figure 2 : Vitesse vs Résolution pour différentes architectures d'ADC

3. Objectif du stage

Ce stage porte sur l'évaluation des performances statiques d'ADCs cycliques. Ce type d'ADC est couramment utilisé dans le développement de capteurs d'image avec un frame-rate élevé. **L'objectif est de déterminer par simulation leur linéarité statique en un temps réduit.**

Dans un second temps, ces travaux pourraient servir à réduire le temps de test de l'ADC. En effet, le test des ADCs dans un circuit électronique représente un temps énorme au regard de la surface de silicium occupée : 1/3 du temps de test total pour seulement 5% de la place occupée ! [1] De plus, le temps de test augmente de manière très importante avec la résolution des ADCs.

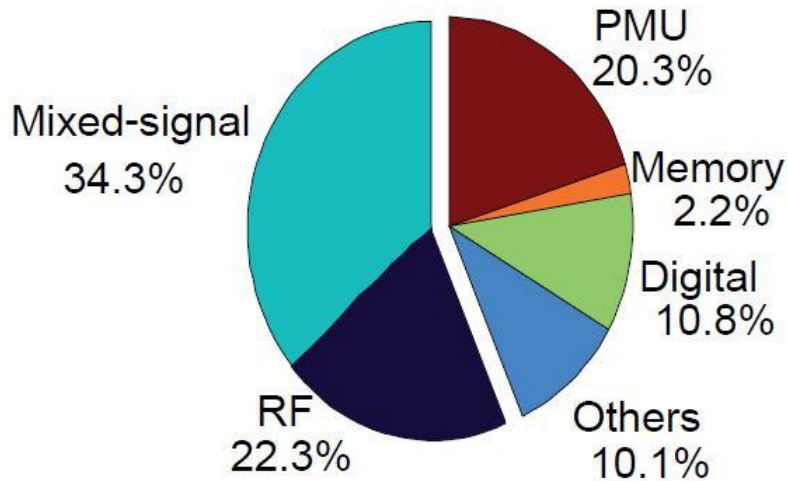


Figure 3 : Répartition du temps de test par bloc [1]

Cette problématique de temps de test est directement liée à des enjeux économiques importants. C'est pourquoi de nombreux travaux de recherche s'attachent à développer de nouvelles techniques de test de linéarité plus performantes, c'est-à-dire plus rapides. Parmi ces travaux, une étudiante de l'Université Grenoble Alpes, Asma Laraba avait comme sujet de thèse, la réduction du temps de test des ADCs pipeline. [1] Mon travail lors de ce stage consiste à :

- Comprendre et adapter la méthode développée par Asma Laraba pour les ADCs pipeline aux ADCs cycliques.
- Développer une autre méthode de réduction du temps de simulation basée sur l'analyse de tensions internes aux ADCs cycliques.

Pour réduire le temps de simulation (test) d'ADCs cycliques, une possibilité est d'injecter en signal d'entrée test de l'ADC un signal de durée réduite par rapport à la méthode classique de l'histogramme. La linéarité statique de l'ADC est alors estimée le plus fidèlement possible à partir de l'information réduite mesurée.

La référence pour le test de linéarité statique est la méthode de l'histogramme. Elle consiste à parcourir la dynamique de l'ADC avec une tension augmentant linéairement de manière pseudo-statique (voir courbe rouge *figure 4*). Pour chaque code de sortie de l'ADC, la plage de tension correspondante est repérée sur l'entrée. C'est cette plage de tension d'entrée qui est appelée « largeur de code ». Cette méthode est certes robuste, mais également très chronophage. C'est pourquoi l'objectif est maintenant d'envoyer une rampe par morceaux présentant de fortes pentes pour ne balayer doucement que les plages de la dynamique d'entrée contenant l'information nécessaire à la reconstruction de la Fonction de Transfert du MDAC (courbe bleue *figure 4, fonctionnement du MDAC expliqué section 5*). A partir de cette information partielle et de l'étude du fonctionnement de l'ADC, une estimation de la fonction de transfert du MDAC et donc des performances statiques de l'ADC est possible.

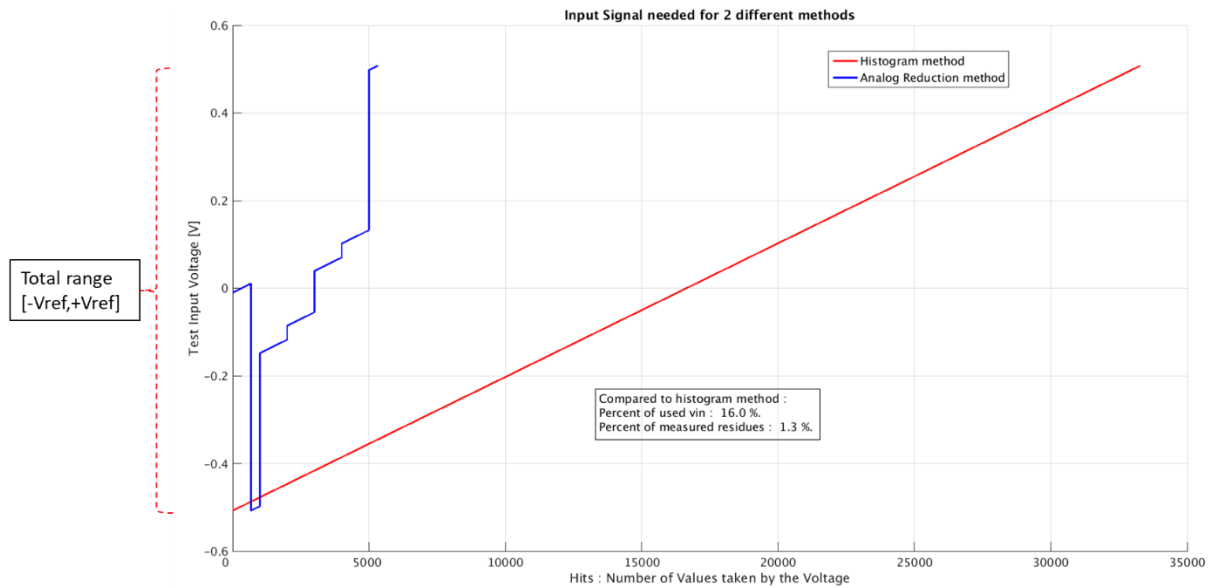


Figure 4 : Objectif = rampe d'excursion temporelle fortement réduite par rapport à la méthode classique

4. Linéarité statique d'un ADC

La linéarité statique d'un ADC est mesurée par rapport à sa fonction de transfert idéale comme illustré sur la figure 5 :

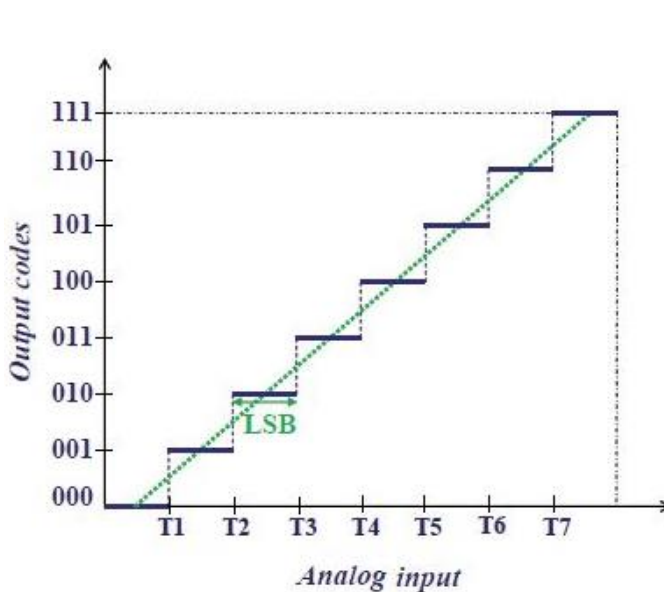


Figure 5 : Fonction de Transfert Idéale & Définition du LSB [1]

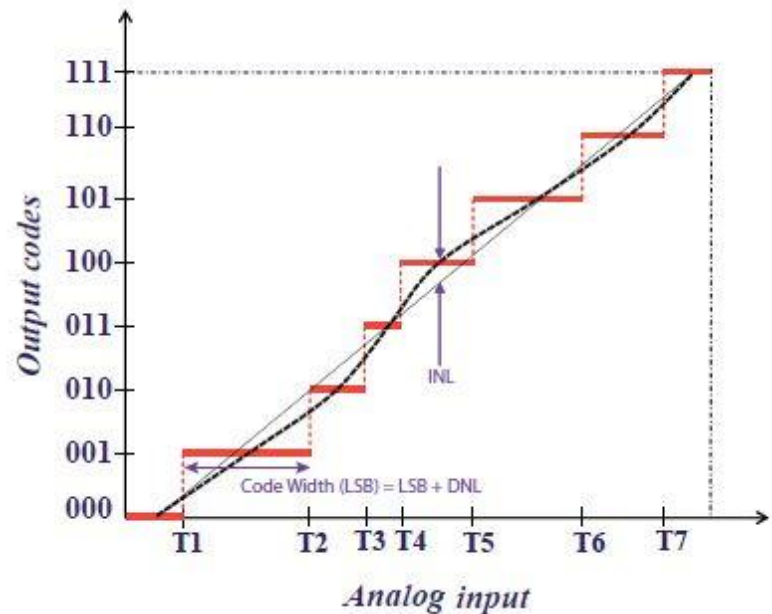


Figure 6 : Fonction de Transfert Réelle, Définitions du DNL et de l'INL [1]

Les deux grandeurs permettant de caractériser la linéarité statique d'un ADC sont (Cypress, 2019 [6]) :

- le DNL - Derivated Non Linearity : Pour un ADC idéal de n bits, la sortie est divisée en 2^n marches de largeurs égales. Cette largeur idéale s'appelle le LSB (figure 5). Toute déviation du LSB est le DNL. Cette grandeur est exprimée en coups, c'est-à-dire en nombre de valeurs d'entrée pour lesquelles la sortie est fausse.

- l'INL - Integrated Non Linearity : Les erreurs de DNL s'accumulent pour former au total l'INL. Cette grandeur est définie comme la déviation maximale mesurée au centre de chaque code entre la pente linéaire idéale de l'ADC et sa caractéristique réelle (*figure 6*). Cette grandeur est également exprimée en coups.

Pour rapidement se rendre compte des performances statiques d'un ADC, il faut tracer DNL & INL [LSB] en fonction des codes de sortie. Des courbes similaires à la *figure 14* sont alors obtenues, où les pics sur le DNL et les sauts sur l'INL sont au niveau des codes de largeurs anormalement éloignées de la largeur idéal = LSB.

5. Fonctionnement d'un ADC cyclique

Un ADC cyclique de N bits échantillonne la tension d'entrée tous les N coups d'horloges. Les performances du MDAC (Multiplying Digital-to-Analog Converter) jouent un rôle important sur les performances de l'ADC. L'architecture simplifiée d'un ADC cyclique est représentée sur la *figure 7*.

- Pour commencer, le signal d'entrée (rebouclé ou l'entrée) est échantillonné
- Ensuite, cette tension est quantifiée par un sous-ADC. Ici, ce sont les deux comparateurs qui remplissent ce rôle et produisent la sortie numérique qui rentre dans le décodeur
- Ce signal est ensuite reconverti en analogue par le sous-DAC.
- Cette tension est soustraite à l'entrée originale pour pouvoir calculer le prochain résidu
- Cette différence de tension est alors amplifiée et éventuellement décalée de $\pm V_{ref}$ pour être ramenée à la dynamique d'entrée de l'ADC et donc du MDAC.

Un ADC cyclique ne comporte qu'un étage rebouclé sur lui-même et traversé au total N fois. Un registre à décalage se charge de récupérer les sorties successives de l'unique étage puis calcule les codes de sortie.

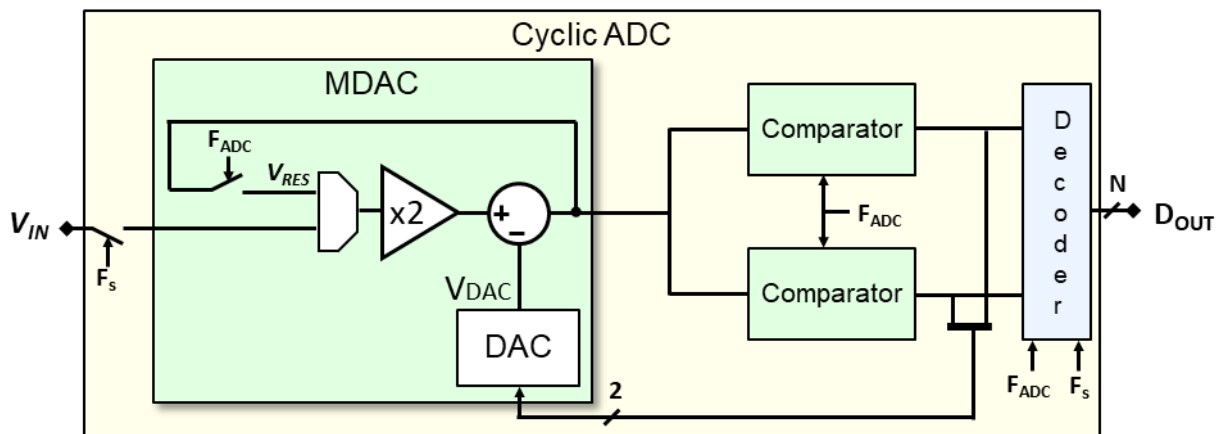


Figure 7 : Architecture d'un ADC cyclique

L'architecture présente est spécifique car elle comporte deux comparateurs, dont les tensions de seuil sont positionnées à 3/8 et 5/8 de la dynamique d'entrée. Une architecture classique utilise un seul comparateur dont la tension de seuil est à la moitié de la dynamique de l'ADC. Cette architecture classique présente l'inconvénient d'être sensible à l'offset du comparateur. L'usage de deux

comparateurs allège la contrainte sur les comparateurs. La *figure 8* représente la fonction de transfert du MDAC pour une architecture à deux comparateurs.

La tension résiduelle $V_{res(i+1)}$ est obtenue à la fin de chaque étape i par la formule :

$$V_{res(i+1)} = 2 * V_{res(i)} + V_{DAC}(i)$$

Où V_{DAC} dépend de la position de $V_{res(i)}$ par rapport aux seuils des 2 comparateurs :

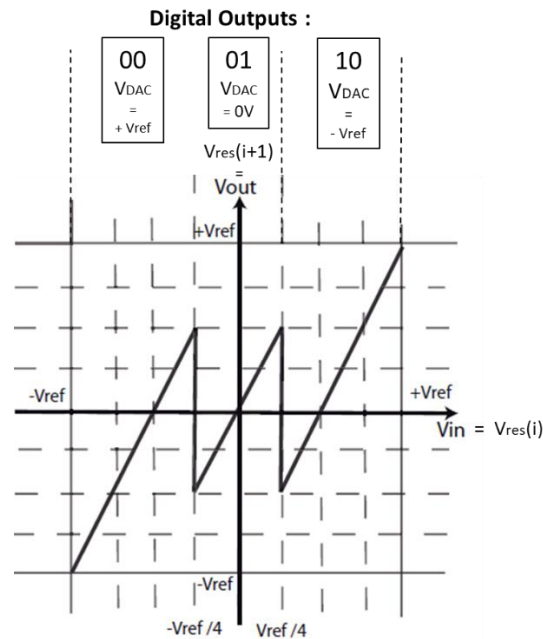


Figure 5 : Fonction de transfert de l'ADC cyclique entre 2 étapes de conversion

Les codes de sortie obtenus sont représentatifs de la valeur de tension résiduelle. A la fin de la conversion une addition numérique entre les bits obtenus permet de reconstruire un code de sortie de N bits.

6. Description du modèle

Pour effectuer les travaux, un modèle comportemental décrit dans l'environnement Matlab/Simulink est mis à disposition par l'entreprise. Ce modèle inclut notamment les paramètres suivants :

- Limites de la dynamique d'entrée
- Tension de référence V_{ref} et bruit associé
- Offset de l'amplificateur opérationnel et bruit associé
- Niveaux des seuils des 2 comparateurs et bruits associés
- Taille des capacités du MDAC
- Bruit en kTC

L'interface permettant de modifier les paramètres du modèle est en *annexe 1*.

Les résultats de simulation présentés dans ce rapport sont obtenus à partir de ce modèle. Les performances obtenues servent de référence pour pouvoir comparer l'estimation des performances d'un ADC cyclique défini par les paramètres comportementaux.

II. Reconstruction analogique

1. Principe

L'objectif de cette méthode est de reconstruire uniquement la fonction de transfert du MDAC pour modéliser la fonction de transfert de l'ADC et obtenir ainsi ses performances statiques. La *figure 9* représente la fonction de transfert d'un MDAC construite à partir des équations représentées sur la *figure 10*. Les paramètres nécessaires à la description de la fonction de transfert réel sont les gains G_{in} et G_{dac} ainsi que la tension d'offset V_{off} . V_{ref} est considérée comme constante et définie.

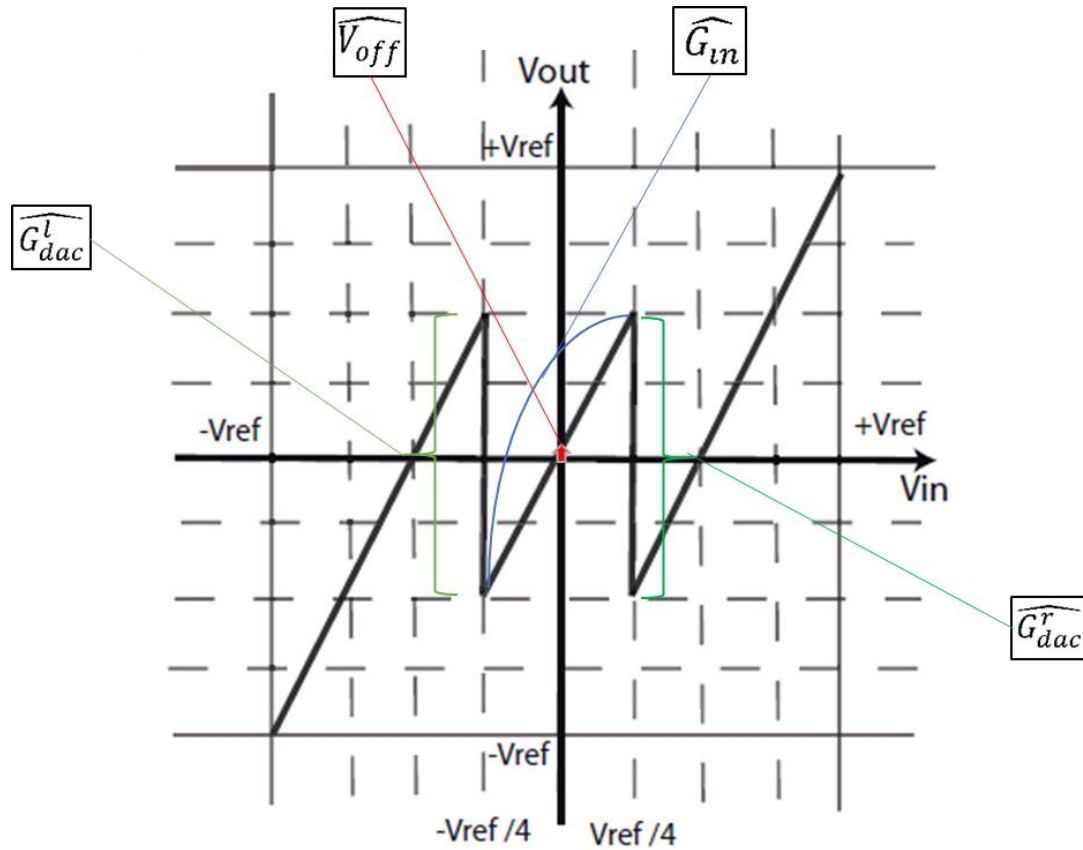


Figure 6 : Fonction de transfert de l'ADC cyclique

$$V_{res}(i+1) = \begin{cases} G_{in} * V_{res}(i) + G_{dac}^l * V_{ref} + V_{off}, & V_{res}(i) < -\frac{V_{ref}}{4} \\ G_{in} * V_{res}(i) + V_{off} & , \quad -\frac{V_{ref}}{4} \leq V_{res}(i) \leq +\frac{V_{ref}}{4} \\ G_{in} * V_{res}(i) - G_{dac}^r * V_{ref} + V_{off}, & V_{res}(i) > +\frac{V_{ref}}{4} \end{cases}$$

Figure 7 : Equations associées au fonctionnement de l'ADC cyclique

2. Mise en place de la méthode

- L'algorithme commence par estimer la tension d'offset V_{off} . Cette tension est commune à toutes les étapes de conversion. Pour cela, une tension nulle est envoyée en entrée de l'ADC et le résidu correspondant est relevé en sortie de la première étape de conversion. Ce dernier est directement la tension d'offset recherchée, voir *figure 4*.
- Une fois l'offset repéré, des intervalles de recherche autour des positions théoriques des transitions centrales des deux premières étapes de conversion peuvent être formés, intervalles translatés de l'offset estimé.
- La sortie numérique pour la 1^{ère} étape de conversion (respectivement la 2^{ème} étape) est alors lue au niveau des intervalles de recherche correspondants. De cette manière la position précise des transitions centrales de l'étape de conversion considérée est repérée, c'est-à-dire la tension pour lesquelles elles ont lieu.
- Il est maintenant possible d'accéder à la valeur de G_{dac} à gauche (respectivement à droite) en mesurant les différences de tension entre les résidus avant et après la transition gauche (respectivement droite). Il est important de préciser que, comme pour le gain G_{in} , ces gains doivent être repérés pour le premier ET le second résidu car ils sont différents en fonction de la parité de l'étape de conversion.
- La valeur de G_{in} pour les deux premières étapes de conversion est directement trouvée au niveau de la pente sur le morceau de droite central entre les deux transitions d'étape.

L'objectif étant de réduire la quantité d'information exploitée pour reconstruire la fonction de transfert, les gains caractéristiques G_{in} , G_{dac} et V_{off} sont déterminés uniquement à partir des deux premiers résidus. Ces deux premières étapes de conversion suffisent à estimer correctement les paramètres de simulation du modèle Matlab utilisés pour comparer les fonctions de transfert de l'ADC reconstruit et l'ADC de référence (modèle comportemental).

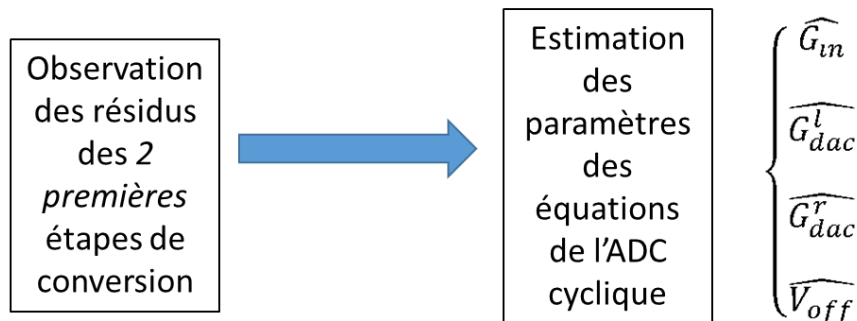


Figure 11 : Principe de la méthode analogique

Tous les gains nécessaires à la simulation des étapes de conversion de l'ADC à partir d'une rampe virtuelle ont ainsi été estimés. La *figure 12* représente le résultat de la simulation de la fonction de transfert de l'ADC reconstruite avec la méthode proposée ainsi que le résultat de la simulation du modèle comportemental caractérisé par les mêmes valeurs de paramètres (gain et offset). Cette première méthode résiste aux erreurs de position (offset, position des tensions de seuil, valeurs des capacités). L'agrandissement au niveau d'un saut de code montre bien que les résultats entre les deux simulations sont très proches. La précision dans l'estimation de la fonction de transfert est abordée dans la section résultats.

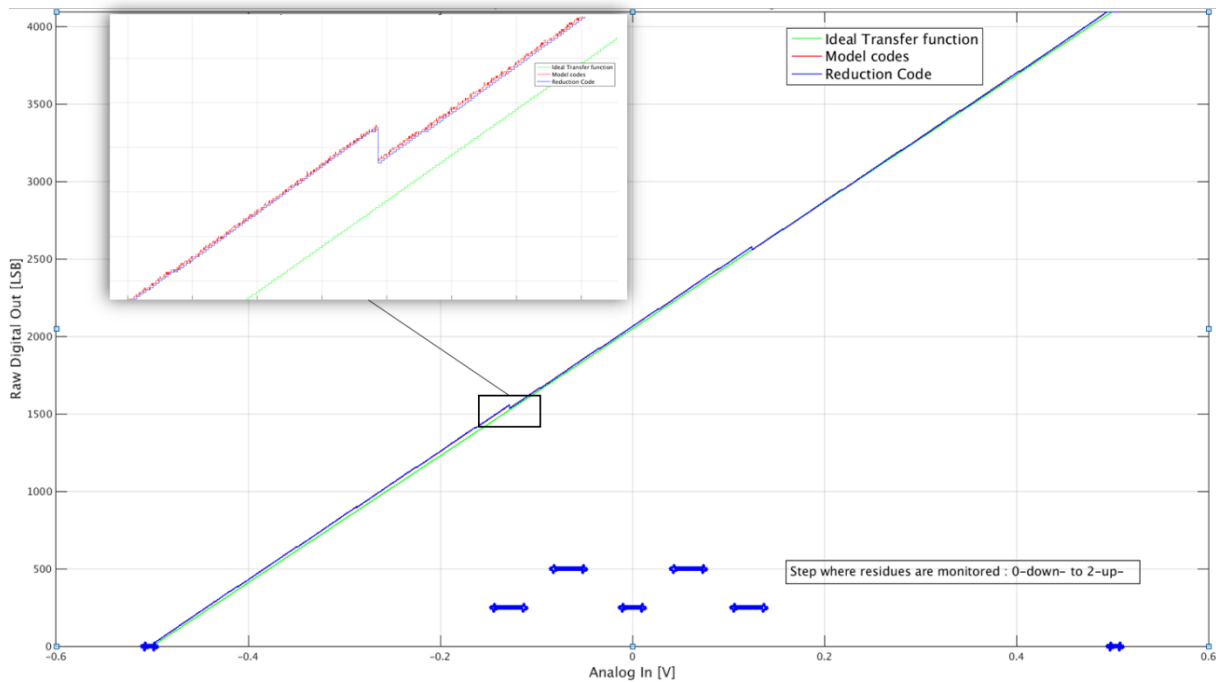


Figure 812 : Fonctions de Transfert avec Offset OTA = 4mV, taille des C = 1fF, seuils décalés

Sur cette figure apparaît également en bleu les intervalles où les résidus sont lus aux différentes étapes de conversion :

- A l'étape 1 au niveau de la tension nulle pour repérer l'offset de l'ADC
- A l'étape 0 (i.e. la tension d'entrée) aux extrémités de la dynamique d'entrée pour pouvoir reconstruire linéairement la rampe virtuelle nécessaire
- Aux étapes 1 et 2 aux niveaux des positions théoriques des transitions traduites de V_{off} .

3. Prise en compte du bruit

Cependant, cette méthode n'est pas robuste au bruit des composants électroniques. Le phénomène prédominant est une variation de la tension de seuil des comparateurs. Cette variation transforme une transition pour une tension fixe en un lot de transitions proches de la tension de seuil attendue. Sur la *figure 13*, la transition unique attendue pour la première étape à $V_{ref}/4$ est remplacée par un lot de transition. Les bruits modélisés sont ceux sur les tensions de référence de l'ADC, sur les seuils des comparateurs et le bruit en kTC. Une étape de « débruitage » est ajoutée à cette méthode d'estimation de gains pour améliorer sa précision face au bruit.

La stratégie adoptée pour s'affranchir du bruit des transitions peut être décomposée de la manière suivante :

- Pour chaque étape de comparaison de l'ADC, des intervalles de recherche sont formés autour des transitions théoriques attendues.
- Une fois ces intervalles formés, la sortie numérique pour l'étape de conversion correspondante est observée et toutes les transitions se produisant dans chaque intervalle sont enregistrées.
- Pour chaque intervalle, seule la transition médiane du lot repéré est gardée.

De cette manière, le nombre théorique de transitions correctement identifiées pour chaque étape de conversion est retrouvé.

La localisation de la tension de seuil médiane de chacun des comparateurs permet d'appliquer la méthode d'estimation de la fonction de transfert du MDAC en s'affranchissant des problèmes de reconstruction liés aux bruits modélisés dans la simulation. La tension médiane est utilisée comme paramètre dans le calcul des gains et de la tension d'offset.

En pratique, seules les transitions observées lors des deux premières étapes sont débruitées car elles contiennent toute l'information nécessaire à la reconstruction de la fonction de transfert du MDAC, c'est-à-dire les différents gains détaillés dans les deux sections précédentes. Pour la première étape, les deux intervalles de recherche sont $[-V_{ref};0]$ et $[0;+V_{ref}]$. Pour la seconde étape, ils sont déterminés en fonction des positions trouvées lors de l'étape 1 (voir *figure 13*).

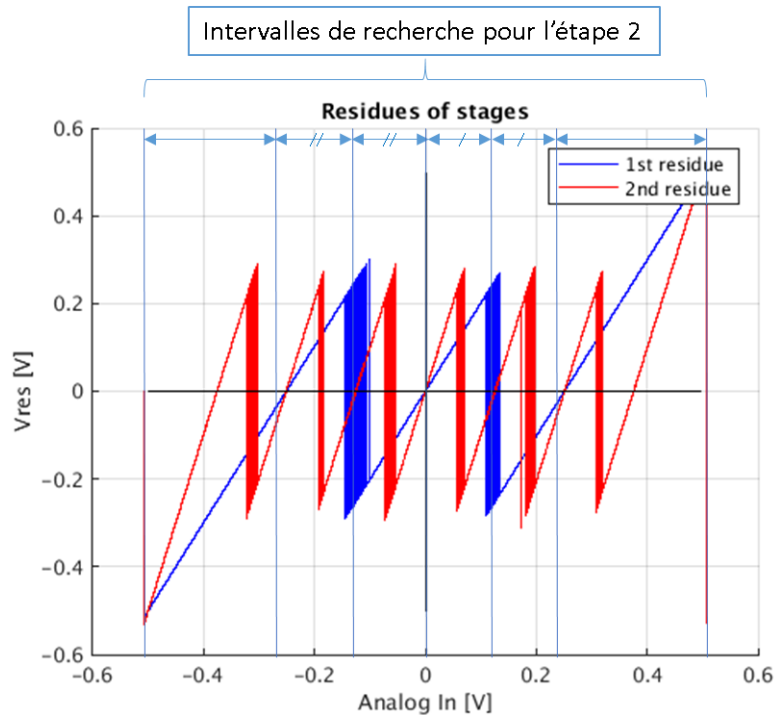


Figure 93 : 1er et 2nd résidus avec bruit sur la position des comparateurs

4. Résultats

La *figure 14* représente les performances de l'ADC obtenues par simulation du modèle comportemental (courbe rouge) et les performances estimées à partir de la méthode de reconstruction analogique proposée (courbe bleue).

La comparaison se fait toujours entre les performances d'un modèle simulé (qui sert de référence) et l'estimation de ces performances par la méthode mise en place.

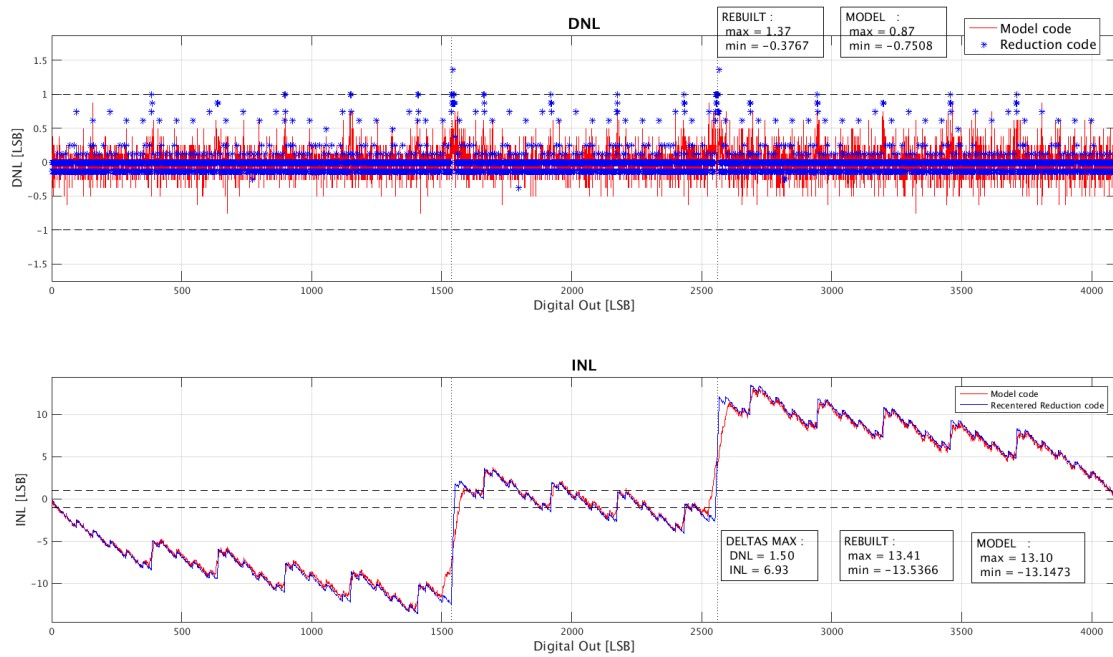


Figure 1014 : DNL & INL - Bruit sur la position des seuils = 6 mV , Bruit sur la réf. & l'offset = 5e-5V , Offset = 4mV
- Taille C = 10fF 0,5-0,9% d'erreur

Sur cette simulation d'un modèle bruité, la courbe bleue représente l'INL de l'ADC reconstruit avec l'algorithme et la courbe rouge, l'INL obtenu par la simulation du modèle bruité. La courbe bleue suit très bien l'INL du modèle rouge. La légère erreur entre les deux courbes est négligeable pour caractériser de manière globale la linéarité statique de l'ADC. Les INL maximum et minimum du modèle sont bien approchés.

L'erreur maximale entre les deux courbes d'INL se trouve au niveau des transitions très bruitées de la 1ere étape de conversion (à environ 3/8 et 5/8 de la dynamique d'entrée, voir les courbes de DNL) qui ont été fortement débruitées en remplaçant les lots de transitions par des transitions uniques, d'où la transition nette reconstruite au lieu de la transition plus progressive du modèle.

La rampe utilisée pour obtenir cette reconstruction est en bleu :

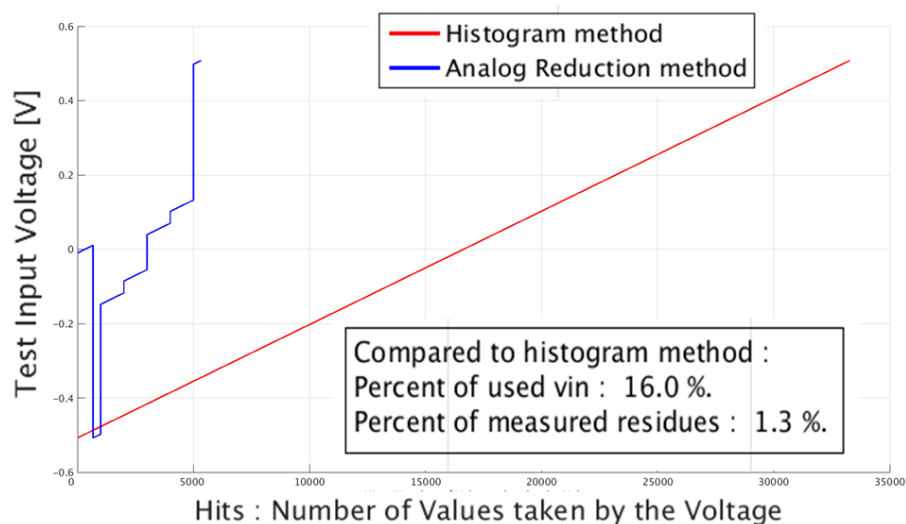


Figure 15 : Signaux d'entrée utilisés en fonction de la méthode choisie

Une réduction significative de la durée du signal d'entrée utilisé pour tester l'ADC est constatée. En effet, Il ne fait plus que 16% de la durée de la rampe classiquement utilisée couvrant toute la dynamique. Si seulement les résidus effectivement utilisés sont considérés, ce nombre est encore divisé par $N=12$ pour obtenir le pourcentage de résidus exploités par rapport au nombre total de résidus de l'ADC : 1.3%.

Les valeurs prises par les paliers de la rampe rapide sont maintenant claires :

- Commencer par repérer l'offset au niveau de la tension d'entrée nulle
- Envoyer en entrée les limites de la dynamique de l'ADC prenant en compte l'offset pour pouvoir construire la rampe virtuelle nécessaire pour appliquer toutes les étapes de conversion.
- Repérer précisément la position des transitions centrales des deux premières étapes entre ces deux limites.

Les paliers créés ici sont pris larges par commodité mais pourraient être réduits pour optimiser la réduction, l'offset étant déjà pris en compte. Attention, les intervalles doivent cependant être pris assez larges pour repérer de gros sauts de code par exemple, ou d'autres erreurs de position.

Simulation Monte-Carlo

La robustesse de la méthode développée est testée par une simulation Monte-Carlo composée de 300 runs bruités. La *figure 16* regroupe :

- L'histogramme des écarts maximaux d'INL entre le modèle simulé et la fonction de transfert reconstruite
- Le résultat de simulation d'un des runs Monte Carlo.

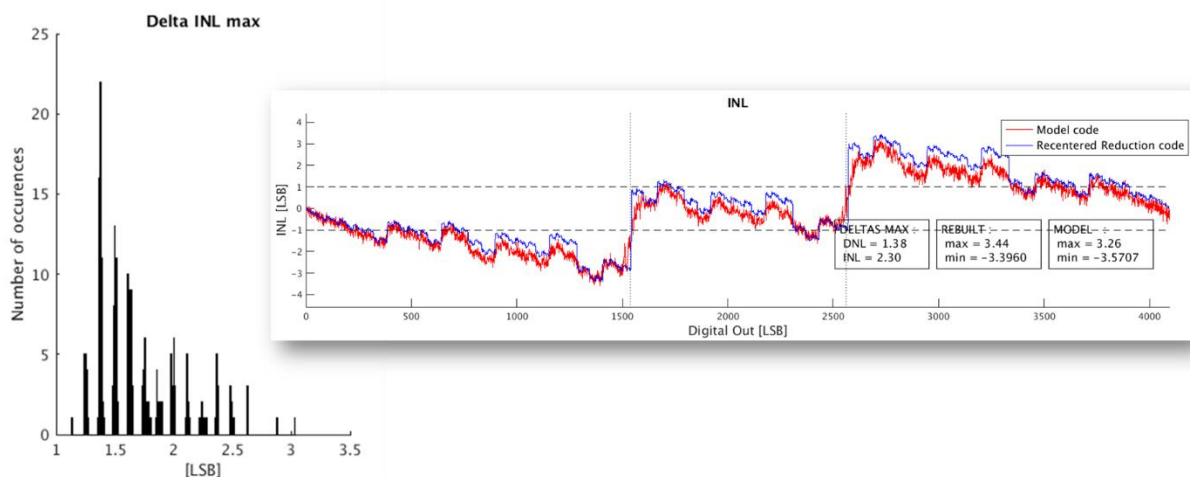


Figure 116 : Histogramme des erreurs d' INL maximales pour 300 simulations aléatoires et un des runs.

La courbe d'INL représentée suit bien le modèle malgré le bruit aléatoire injecté. Cependant, le niveau de bruit est ici important : La courbe d'INL du modèle sort de la fourchette ± 1 LSB. Avec un modèle moins bruité, il faut faire plusieurs simulations du même run pour atteindre une mesure précise. D'autre part, l'erreur maximale d'INL moyenne est d'environ 1.5 LSB, ce qui est très correcte avec la grande réduction d'information de départ.

5. Architecture du code

La *figure 17* représente les étapes de reconstruction enchainées dans le code Matlab, reprises ici sous forme de schéma bloc :

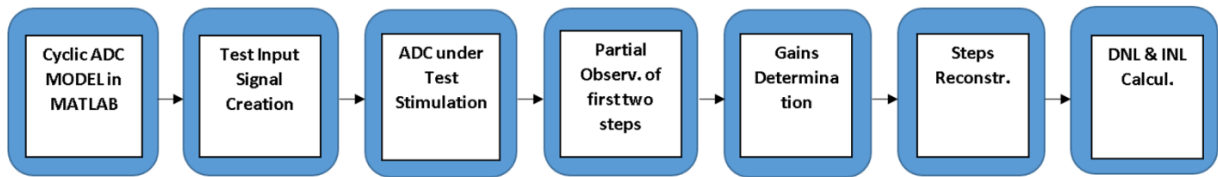


Figure 127 : Architecture du code

Chaque bloc correspondant à une fonction ou une section clairement définie dans le code.

L'inconvénient de cette méthode est la nécessité de connaître précisément la tension de résidu en sortie des deux premières étapes de comparaison de l'ADC. Dans un environnement de test réel, ces tensions sont inaccessibles. Mais l'estimation des gains de la fonction de transfert du MDAC en utilisant uniquement les codes de sortie des comparateurs n'a pas été concluante car la précision des résultats était dégradée par l'influence du niveau des seuils de comparaison.

Une nouvelle méthode basée sur la détection des transitions des comparateurs répond à la contrainte d'estimer les performances d'un ADC cyclique en ayant accès uniquement aux codes numériques de sortie et à la tension d'entrée. Cette méthode est décrite dans le paragraphe suivant.

III. Reconstruction numérique

1. Principe

L'étude de l'architecture d'un ADC cyclique montre que les transitions entre les codes de sortie proviennent du basculement d'un comparateur à une étape donnée de la conversion. La reconstruction de la fonction de transfert de ce type d'ADC revient à détecter les tensions de seuil des comparateurs pour chaque étape de la conversion et de créer une table d'association entre le couple étape de la conversion/comparateur et les transitions de sortie :

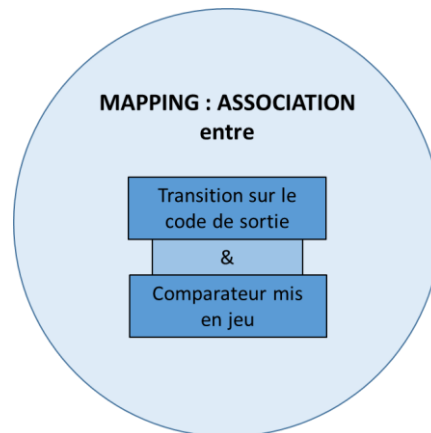


Figure 18 : Etape du Mapping

La création de cette table d'association s'effectue en sélectionnant seulement les transitions sur le code de sortie représentatives du changement d'état d'un comparateur à une étape de conversion donnée. Ce mapping est détaillé dans la section 3 de ce chapitre. Enfin, la largeur des codes de sortie autour des transitions sélectionnées est mesurée et la reconstruction de la fonction de transfert s'effectue à partir de la répétition de l'emplacement des transitions et de la largeur du code associée sur toute la dynamique de l'ADC. Ce simple « copier-coller » est représenté sur la *figure 19*.

Pour reconstruire la fonction de transfert de ce type d'ADC, il faut repérer les changements d'état des comparateurs pour chaque étape de la conversion. Le cas d'étude est un convertisseur cyclique 12 bits à un étage de 1.5 bits (soit 2 comparateurs). Pour effectuer le mapping, il est donc nécessaire de déterminer l'emplacement de 24 transitions correspondant au changement d'état des deux comparateurs pour chacune des 12 étapes de la conversion.

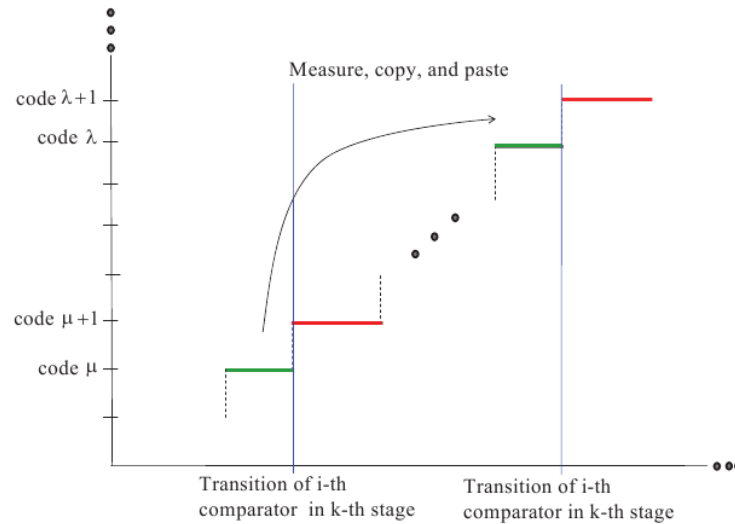


Figure 19 : Principe de la méthode numérique [1]

2. Architecture du code

La mise en œuvre de cette méthode se divise en deux phases :

- Une phase de **Training** sur une petite population d'ADCs. Pendant cette phase, le Mapping s'effectue en enregistrant la sortie numérique pour toutes les étapes de conversion sur toute la dynamique de l'ADC et les codes de sortie correspondants.
- Une phase d'**Application** pour les ADCs à tester. Pendant cette phase, les données mesurées sont bien moindres car elles correspondent uniquement à un échantillon des codes de sortie. C'est à partir de cet échantillon et du mapping que les performances des ADCs sous test sont estimées.

La *figure 20* reprend en détail chaque étape de la reconstruction numérique de la fonction de transfert d'un ADC cyclique.

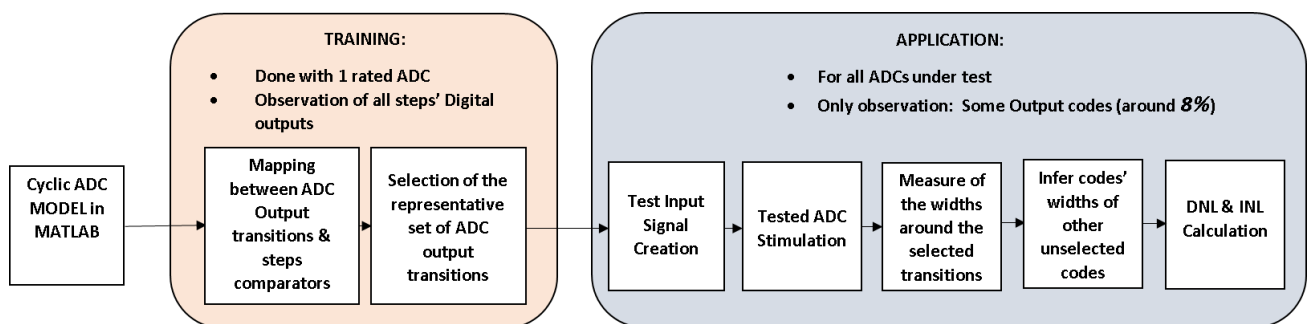


Figure 20 : Architecture du code

Lors de la phase d'application, il faut :

- Commencer par repérer l'offset pour former des intervalles centrés autour des transitions sur le code de sortie visées contenues dans le set de training. La suite du signal d'entrée adapté à l'ADC testé est ensuite générée.

- Stimuler l'ADC avec le signal d'entrée adapté
- Mesurer les largeurs de codes autour des transitions sélectionnées
- Etendre les mesures autour des transitions sur le code de sortie non sélectionnées
- Calculer les performances statiques : DNL & INL.

3. Retour sur le Mapping

Transitions forcées et naturelles d'une étape

Pour bien comprendre l'étape de Mapping, cette distinction est nécessaire. Une transition naturelle d'une étape de conversion donnée s est due au franchissement d'un des deux comparateurs de cette étape s . En effet, lorsque la sortie numérique pour l'étape de conversion s est observée, les transitions peuvent avoir deux sources :

- Soit le basculement d'un comparateur de l'étape de conversion s
- Soit le basculement d'un comparateur d'une étape de conversion antérieure.

La figure 21 représente un graphe des deux premiers résidus d'un convertisseur 12 bits -1.5 bits par étape- qui illustre cette distinction :

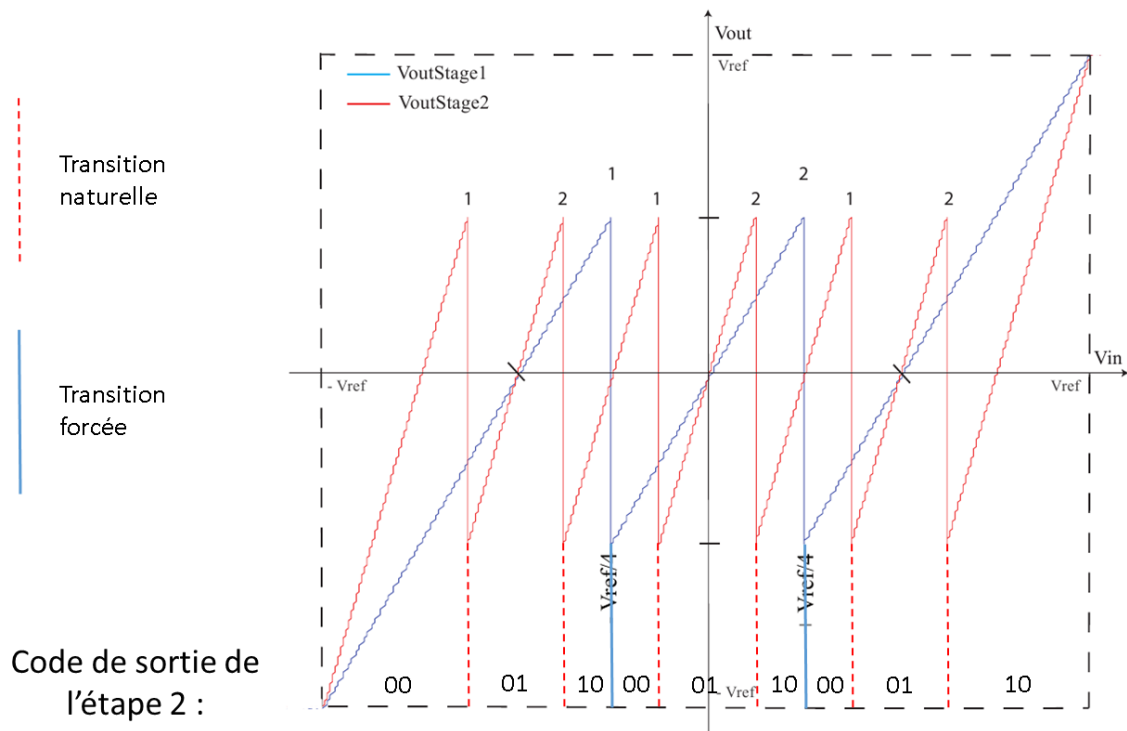


Figure 21 : Transitions forcées et naturelles de l'étape 2

Pour effectuer le Mapping, la sortie numérique est enregistrée pour toutes les étapes de conversion. Pour pouvoir proprement caractériser les comparateurs d'une étape de conversion donnée, uniquement les transitions naturelles sont retenues sur la sortie numérique pour l'étape de conversion considérée, c'est à dire (00→01, comparateur 1) et (01→10, comparateur 2).

Association Transition sur code de sortie – comparateur

Pour chaque transition naturelle repérée sur la sortie numérique entre les instants i et $i+1$:

- La sortie numérique pour toutes les autres étapes la chaîne de conversion en cours à l'instant i est analysée. De cette manière, s'il y a eu d'autres basculements de comparateurs dans cette chaîne de conversion, ils sont repérés.
- La transition correspondante sur le code de sortie de l'ADC est calculée à partir de la sortie numérique enregistrée tout au long de la chaîne de conversion.
- A une transition sur le code de sortie de l'ADC est toujours lié le premier comparateur dans la chaîne de conversion d'une tension d'entrée présentant une transition naturelle.
En effet, il y a souvent plusieurs manières d'obtenir un code de sortie (possible à cause de l'overlap entre sorties numériques des différentes étapes de conversion lors du calcul du code final et du bruit), donc différents comparateurs candidats pour représenter une transition sur le code de sortie. Le comparateur intervenant dans l'étape de conversion arrivant la plus tôt est sélectionné.

La *figure 22* est un exemple de résultats de simulation permettant de définir à quel comparateur associer la transition entre les codes 88 et 89.

Le comparateur C1 change d'état à l'étape 5. C'est un candidat pour être associé à cette transition sur le code de sortie. En effet, selon ce résultat de simulation, aucune transition n'est repérée pendant les étapes de conversion 1 à 4 (mention No_Tr écrite en sortie). Il sera effectivement associé à la transition 89 s'il n'y a pas de comparateur, intervenant plus tôt dans la chaîne de conversion, qui produit également cette transition pour une entrée différente.

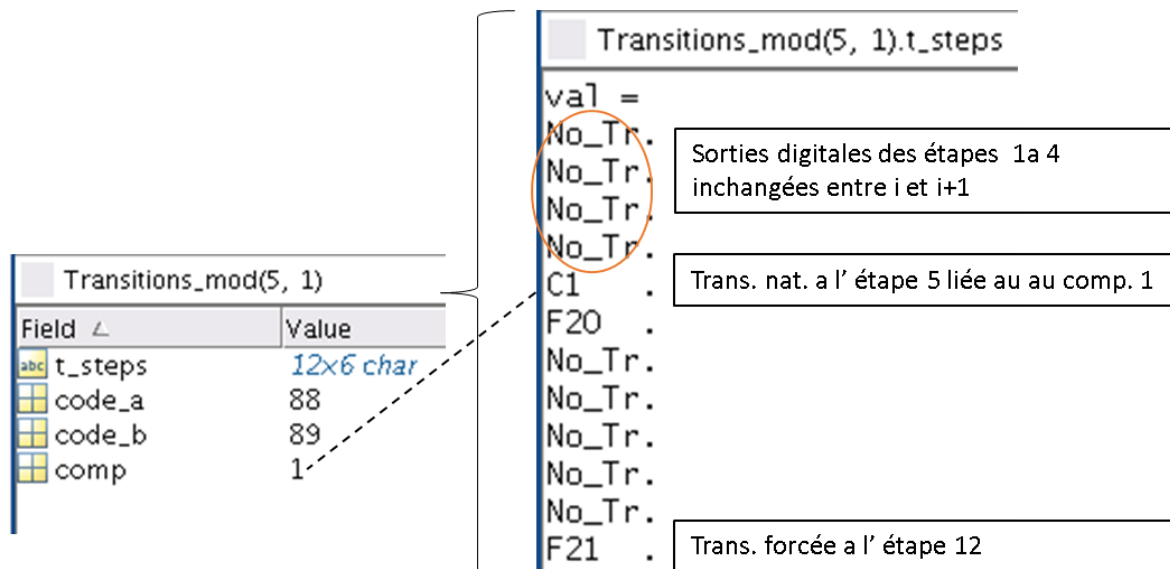
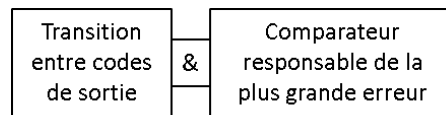


Figure 1322 : Exemple de transition naturelle : la 1ère repérée à l'étape de conversion 5

Cette étude est terminée lorsque toutes les transitions étudiées permettent de reconstruire la fonction de transfert de l'ADC. Le résultat de cette étude est une table d'association entre les transitions des codes de sortie et les comparateurs responsables de la plus grande erreur.



ADC_Transitions_mod

4095x3 double

	1	2
1539	10	1
1540	8	2
1541	10	2
1542	1	1
1543	10	1
1544	7	2

Figure 1423 : Table d'association

4. Sélection des transitions

Les transitions sont sélectionnées comme la distribution théorique MAIS de manière automatique. En effet, cela est nécessaire pour que la mesure des codes associés à une transition d'une étape donnée ne soit pas faussée par l'influence/l'erreur sur la position d'un comparateur appartenant à une étape précédente. Ainsi, la sélection se fait en partant des limites de la dynamique d'entrée :

- A gauche pour les comparateurs 1
- A droite pour les comparateurs 2.

Figure 24 : - Haut : Résidus des 2 premières étapes de conversion

- Bas : Distribution idéale et sélection des transitions pour les 3 premières étapes [1]

D'autre part, chaque code de sortie est formé par deux transitions, gauche et droite. La transition issue de l'étape la plus petite entre les deux lui est attribuée. Les transitions sélectionnées sont ainsi entourées d'au moins 2 codes dont elles sont responsables de la largeur.

5. Création du signal d'entrée

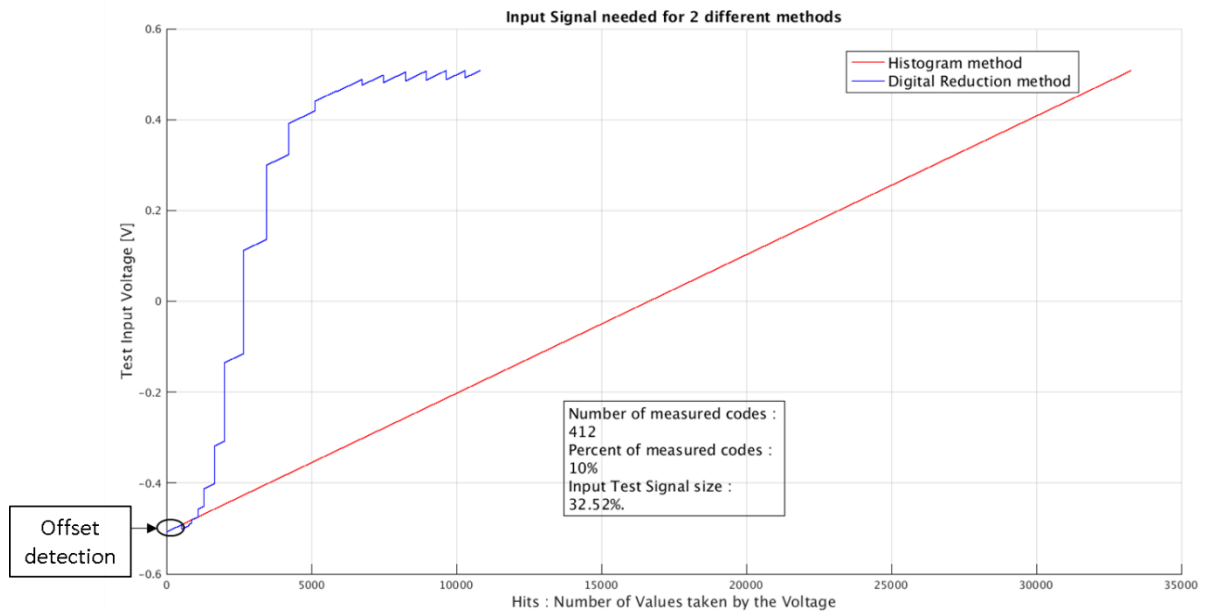


Figure 25 : Signaux d'entrée utilisés en fonction de la méthode choisie

- La détection de l'offset nécessaire pour former le bon signal d'entrée est retrouvée dans cette méthode.
- La disposition des transitions sélectionnées est remarquable : très écartées au centre et accumulées aux limites de la dynamique
- Les intervalles ont été pris ici larges, et pourtant une diminution de 2/3 de la durée du signal d'entrée nécessaire par rapport à la méthode avec histogramme est obtenue. Seuls 10% des codes de l'ADC sont mesurés.
- Plus son étape associée est précoce, plus le nombre de codes mesurés autour de la transition associée est important car l'erreur sur sa position est plus grande.

6. Mesure des largeurs de codes

Les transitions visées sont contenues dans l'échantillon formé pendant la phase de training. Avec cette seule connaissance, des intervalles de recherche centrés autour des positions idéales sont créés. Cependant, une fois l'offset repéré, ces intervalles sont translatés pour ne pas chercher à cote des codes visés.

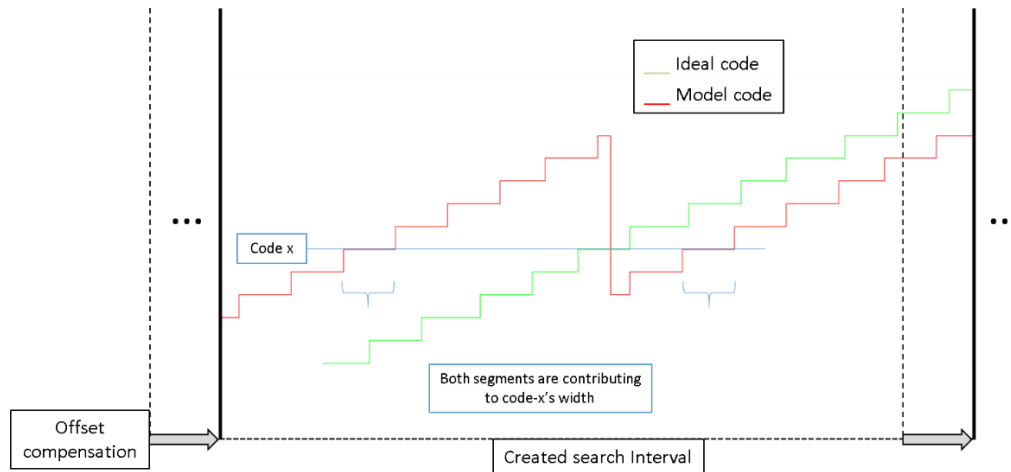


Figure 26 : Formation des intervalles de recherche

Une fois l'intervalle formé, il faut trier les codes qu'il contient. En effet, les mismatches des valeurs de capacités provoquent des sauts de code comme en rouge sur la *figure 26*. Or, la fonction qui mesure ensuite les largeurs de code opère de manière linéaire : le tri de l'intervalle est donc nécessaire avant la mesure.

7. Résultats

La question du Training à utiliser est très importante. Le but étant qu'il soit fait avec un ADC le plus proche possible de ceux qui vont être testés rapidement par la suite. Il peut donc être fait avec des bruits moyens et un ADC utilisé dans les conditions nominales. Cependant, si les ADC testés sont très différents les uns des autres, un training fait avec un ADC presque parfait (aucun bruit, gains légèrement différents) se justifie. C'est en effet ce dernier qui donne le plus de résultats cohérents avec des ADCs atypiques.

Le training suivant est choisi :

Training utilisé:

- Erreur valeur des capacités = 0,1 – 0,2%
- Aucun bruit

Voici les résultats obtenus pour deux applications :

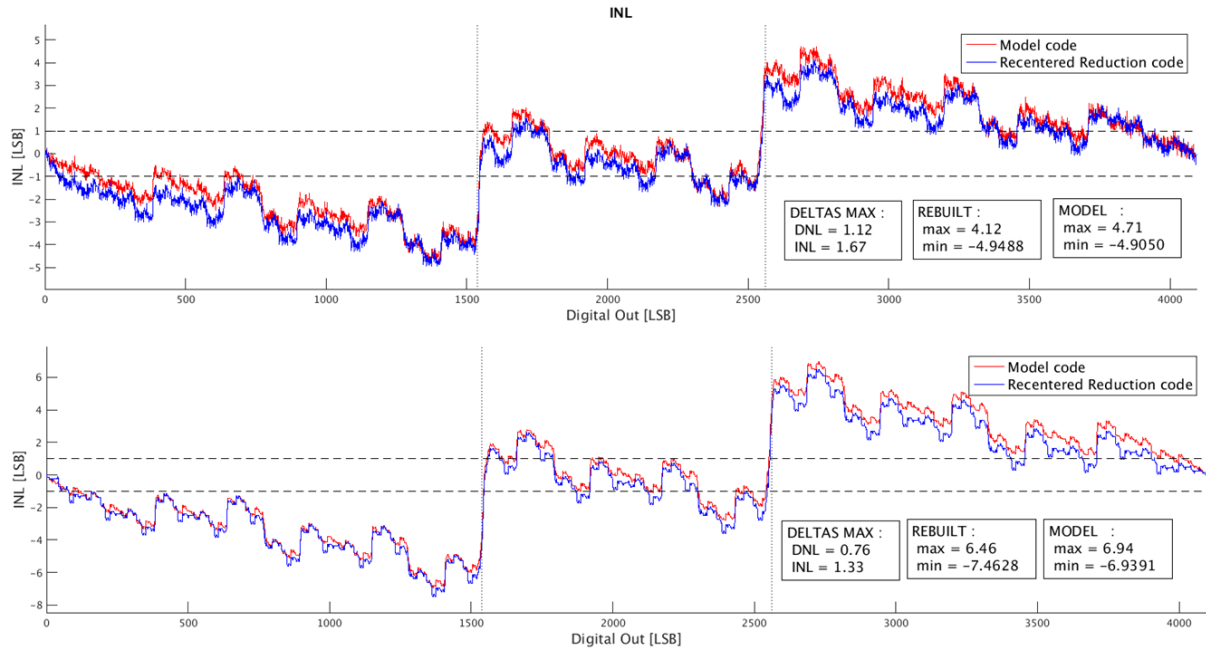


Figure 27 : INL pour 2 applications avec modèle: - (haut) fortement bruité - (bas) légèrement bruité

Dans les deux cas l'approximation faite est cohérente. Visuellement, la courbe bleue reconstruite suit correctement la courbe rouge du modèle. Les valeurs des erreurs maximales d'estimation de DNL et d'INL confirment cette observation. Les valeurs extrêmes de DNL et d'INL sont correctement estimées.

Enfin, une première validation de la robustesse du modèle est réalisée avec une simulation Monte-Carlo de 250 essais bruités. L'histogramme des erreurs maximales sur l'INL est tracé. La moyenne est de 1.2 LSB, ce qui reste très correct :

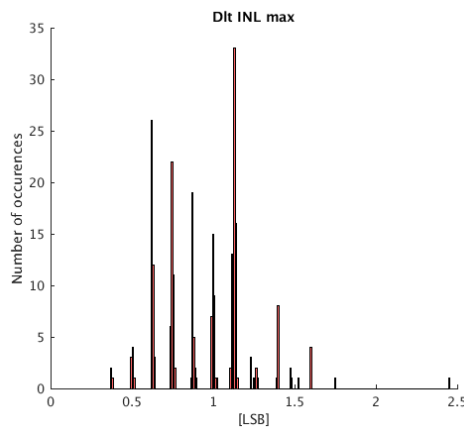


Figure 28 : Histogramme des écarts maximums d'INL pour 250 simulations Monte-Carlo

IV. Conclusion

Pendant ce stage, deux méthodes de test de linéarité statique ont été mises au point pour l'ADC cyclique. Elles diffèrent radicalement d'après les données qu'elles nécessitent :

- La méthode analogique est réservée aux simulations comme elle demande d'avoir accès aux résidus, tensions internes de l'ADC délicates à extraire sur des ADCs réels.
- La méthode numérique pourrait être implantée pour tester de vrais ADCs en plus d'être utile en simulation. En effet, elle n'a besoin en phase d'application que d'un échantillon des codes de sortie.

Pour chaque méthode proposée, la réduction du temps de test (ou de simulation) est très significative (divisé par un facteur 5) en conservant une précision suffisante sur l'estimation des performances.

Les deux méthodes développées ici sont en effet opérationnelles mais non optimisées pour l'instant :

- Les intervalles de recherche sur la fonction de transfert de l'ADC où l'information est extraite peuvent par exemple être réduits
- Le nombre de codes à mesurer autour des transitions sélectionnées peut être optimisé en fonction des étapes de conversion.

Enfin, pour envisager une utilisation industrielle de ces méthodes, il faudrait qu'elles repèrent -ou du moins résistent- aux erreurs dynamiques des ADCs. Mon stage s'est en effet concentré sur la linéarité statique des ADCs. Cela constituait, en conclusion, une première étape nécessaire et intéressante pour d'éventuelles applications futures.

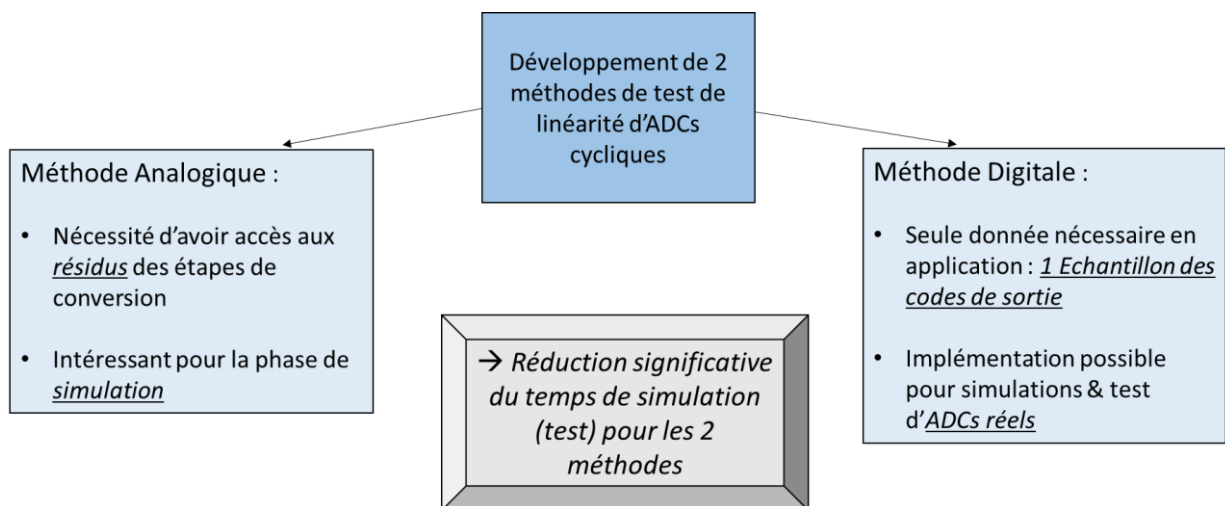


Figure 29 : Conclusion

V. Références

- [1] Asma Laraba. *Conception en vue de test de convertisseurs de signal analogique-numérique de type pipeline.*. Autre. Université de Grenoble, 2013. Français. <NNT : 2013GRENT040>. <tel-00947360>
- [2] Puppala, A. (2012). *Design of a Low Power Cyclic/Algorithmic Analog-to-Digital Converter in a 130nm CMOS Proches*. [ebook] Linkopings universitet, p.11. Available at: <http://liu.diva-portal.org/smash/get/diva2:545838/FULLTEXT01.pdf> [Accessed 3 Jul. 2019]
- [3] Sansen, W. M. (2006). *Analog Design Essentials*. Leuven, Belgium: Springer.
- [4] subratachakTechnology, diary, literature, sports. (2015). *Camera Sensor*. [online] Available at: <https://subratachak.wordpress.com/2015/12/13/camera-sensor/> [Accessed 5 Sep. 2019].
- [5] Cypress. (2019). *Definitions of INL and DNL in an ADC*. [online] Available at: <https://community.cypress.com/docs/DOC-12098> [Accessed 13 Feb. 2013].
- [6] KLEDROWETZ, V. and HAZE, J. (2019). *Basic Block of Pipelined ADC Design Requirements*. [online] radioeng.cz. Available at: <http://Basic Block of Pipelined ADC Design Requirements> [Accessed 6 Sep. 2019].

VI. Annexes

1.

Menuparametres

Parametres systeme

ADC :

Resolution :
 Bits

Frequence d'echantillonnage :
 Hz

Dynamique d'entree :
 ; V

Quantum :
 DAC V/LSB

Reference :
 Vref V

Bruit sur la reference :
 V ☐ Enable

OTA :

Vss : V Vdd : V

Gain en Boucle Ouverte :
 dB ☐ Fini

Bande passante unitaire :
 Hz

Slew Rate :
 V/s

Offset :
 V

Bruit de l'OTA :
 V ☐ Enable

COMPARETEURS :

Seuil de comparaison haut :
 V

Seuil de comparaison bas :
 V

Bruit sur le seuil de comparaison haut :
 V ☐ Enable

Bruit sur le seuil de comparaison bas :
 V ☐ Enable

MDAC :

Taille des capacites des l'etage :
 fF

Erreur sur la valeur des capacites :
 Sigma_C fF ☐ Enable

Gain MDAC 1 :
 Gin1

Gain MDAC 2 :
 Gin2

Bruit en kTC : ☐ Enable

Figure 30 : Interface permettant de modifier les paramètres du modèle comportemental

VII. Résumés

1. Français

Pendant ce stage, deux méthodes de test de linéarité statique ont été mises au point sur Matlab pour l'ADC cyclique. Elles diffèrent radicalement d'après les données qu'elles nécessitent :

- La méthode analogique est réservée aux simulations comme elle demande d'avoir accès aux résidus, tensions internes à l'ADC délicates à extraire sur des ADCs réels. Elle a été développée avec l'aide de Matthieu suite à nos observations. Le principe est d'estimer à partir des résidus des deux premières étapes de conversion pour des plages précises de la dynamique d'entrée les gains de la fonction de transfert de l'ADC. Il est alors possible reconstruire toutes les étapes de conversion de l'ADC à partir de ces gains et d'une rampe virtuelle.
- La méthode numérique, elle, pourrait être implantée pour tester de vrais ADCs en plus d'être utile en simulation. En effet, elle n'a besoin en phase d'application que d'un échantillon des codes de sortie. Elle est l'adaptation de la méthode développée par Asma Laraba pour les ADCs pipeline dans sa thèse [1]. Elle s'appuie sur le fait que chaque transition sur le code de sortie de l'ADC présente une erreur de position principalement due à un comparateur à une étape de conversion donnée de l'ADC. En créant une table d'association (transition-comparateur mis en jeu), il devient alors possible de sélectionner qu'un échantillon réduit de codes à mesurer puis étendre ces mesures autour des autres transitions sur le code de sortie. Des transitions sur le code de sortie dues au même comparateur de la même étape de conversion auront en effet les mêmes largeurs de codes autour d'elles.

Pour chaque méthode proposée, la réduction du temps de test est significative et peut être d'un facteur 5 tout en gardant des résultats très fidèles aux modèles testés.

2. Anglais

Two test-linearity methods for cyclic ADCs have been developed on Matlab during this internship. Their large difference is the data they require:

- The analogic method is reserved for simulations since access to residues - intern tensions sensitive to extract - is needed. It was developed with Matthieu's help and results from our observations. The principle is to estimate from the first two residues for specific entries all the characteristic gains of the ADC's Transfer Function. It is then possible to rebuild all conversion steps of the ADC, from a virtual ramp and these very gains.
- The digital method could besides be implemented to test real ADCs. It results from the adaptation of Asma Laraba's thesis on pipeline ADCs [1]. In the application phase, only a reduced set of output codes have to be measured. The method relies on the fact that each ADC output transition position can be linked with a comparator at a step of a conversion process. After having created thus an association table between ADCs' transitions & comparators, only a reduced set of transitions needs to be measured. Finally, these code-width measures are extended around the unselected ADC transitions to rebuild the whole ADC transfer function, depending on the association table. Transitions dues to the same comparator at the same step will have same codes widths around them.

In both cases the test time has been significantly reduced (factor 5) with correct results.