

N° d'ordre : 2996

THESE

Présentée à

L'UNIVERSITE BORDEAUX 1

ECOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGENIEUR

Par Guillaume MONNERIE

Pour obtenir le grade de

DOCTEUR

SPECIALITE : Electronique

Etude et modélisation de sources de bruit dans les structures à temps discret

Soutenue le 8 juillet 2005

Après avis de :

M.	Y. HERVE	Maître de Conférences, PHASE - ENSPS	Rapporteur
M.	P. LOUMEAU	Maître de Conférences, LTCl - ENST Paris	Rapporteur

Devant la commission d'examen formée par :

M.	D. DALLET	Professeur, IXL - ENSEIRB	Examineur
M.	H. GUEGNAUD	Ingénieur, EADS	Examineur
M.	H. LEVI	Professeur, IXL - Université Bordeaux I	Directeur de thèse
M.	A. TOUBOUL	Professeur, IXL - Université Bordeaux I	Président

Membres invités :

Mme.	N. LEWIS	Maître de Conférence, IXL - Université Bordeaux I	Membre invité
M.	M. ROBBE	Ingénieur, EADS	Membre invité

A Emmanuelle pour son soutien

Ce document propose une synthèse de mes travaux de recherche effectués conjointement au sein du laboratoire de microélectronique de l'ENSEIRB, l'IXL, et au sein de l'équipe ASIC d'EADS-TELECOM. Avant de remercier nommément les personnes ayant participé à la réussite de cette petite entreprise, j'aimerais exprimer toute ma gratitude à mes encadrants pour leur présence ainsi que pour leur très grande disponibilité. En fin de compte, la soutenance a été l'unique moment où j'ai été seul.

J'aimerais témoigner ma reconnaissance à Messieurs André TOUBOUL et Jean-Marc MARCZAK pour leur confiance et leur accueil respectif dans le laboratoire IXL et dans l'équipe ASIC d'EADS-TELECOM. Je tiens tout particulièrement à remercier Monsieur André TOUBOUL d'avoir accepté de présider mon jury de thèse.

J'exprime également toute ma gratitude à Monsieur Hervé LEVI pour avoir accepté de prendre la direction de ma thèse et pour le soutien, sans faille, tant moral que scientifique qu'il m'a apporté.

Outre le fait d'avoir participé de près à mes travaux, le point commun de mes encadrants directs a été une grande patience à mon égard. Ainsi j'aimerais remercier Madame Noëlle LEWIS, pour avoir entretenu cette qualité humaine en acceptant d'encadrer, et de cadrer, les avancées de ma thèse. De même, je remercie Monsieur Dominique DALLET pour les nombreux débats contradictoires scientifiques et politiques que nous avons eus tant à l'IXL que durant nos repas. Je remercie également Monsieur Michel ROBBE d'avoir accepté de consacrer une part non négligeable de son temps à me transmettre une partie de son savoir. Il en va de même pour Monsieur Hervé GUEGNAUD qui m'a guidé dans l'univers de la R&D au sein d'EADS. Je leur suis également très reconnaissant d'avoir pris part à mon jury.

Je remercie vivement Messieurs Yannick HERVE, maître de conférences à l'ENSPS-Strasbourg et Patrick LOUMEAU, maître de conférences à l'ENST-Paris, pour l'intérêt qu'ils ont porté à mes travaux en acceptant d'examiner ce mémoire et d'en être les rapporteurs.

Je ne voudrais pas oublier l'équipe ASIC et leur témoigner toute ma sympathie pour leur soutien moral notamment lors du retour du « bootstrap ». Je pense notamment à Monsieur Thierry LAGUTERE qui a partagé les mêmes conditions de thèse à trois bureaux de moi, à Messieurs Sami AISSA, Didier DHERET, Stephan DOUCET et Laurent RENAUD qui ont supporté mes états d'âme, durant les pauses café, les repas ainsi que durant nos trajets de RER. J'aimerais également remercier Monsieur Michel ROBIN, mon colocataire de bureau, pour m'avoir accordé sa confiance. L'équipe ne serait pas complète sans Monsieur Rolland STOFFEL, l'inventeur du concept des « aéroports sous-marins ».

Enfin, j'exprime toute ma sympathie à tous les membres du laboratoire IXL et d'EADS-TELECOM que je n'ai pas nommés et qui ont par leur présence contribué au bon déroulement de mes travaux.

Ma dernière pensée revient à l'émission radiodiffusée « là bas si j'y suis » présentée par Monsieur Daniel MERMET sur France Inter, qui, à travers un regard - partisan ? - sur le monde, m'a permis de relativiser les petites contrariétés de la vie d'un doctorant.

SOMMAIRE

SOMMAIRE

INTRODUCTION GENERALE	17
CHAPITRE I	21
I. INTRODUCTION	23
II. LES SYSTEMES DE RADIOCOMMUNICATION	24
II.1. Chaîne complète d'émission/réception	24
II.2. Les architectures de réception	25
II.2.1. L'architecture hétérodyne	26
II.2.1.1. Le concept	26
II.2.1.2. La réjection d'image	27
II.2.1.3. L'architecture hétérodyne à double changement de fréquence	28
II.2.1.4. L'architecture hétérodyne polyphasée	28
II.2.2. L'architecture homodyne	30
II.2.2.1. Le concept	30
II.2.2.2. L'architecture homodyne polyphasée	31
II.2.3. L'architecture à faible fréquence intermédiaire	31
II.3. L'architecture de réception retenue	32
III. LA CONVERSION ANALOGIQUE/NUMERIQUE UTILISANT LA MODULATION SIGMA DELTA	34
III.1. La conversion analogique/numérique « classique » : généralités	34
III.1.1. L'échantillonneur bloqueur	34
III.1.2. Le quantificateur	35
III.1.3. Le convertisseur analogique/numérique	37
III.2. De la modulation Delta à la modulation Sigma Delta	39
III.3. Les modulateurs Sigma Delta	42
III.3.1. Les modulateurs Sigma Delta passe bas : théorie générale	43

III.3.1.1.	Modulateur du premier ordre.	43
III.3.1.2.	Modulateur du second ordre et généralisation	45
III.3.1.3.	Structure MASH	47
III.3.2.	Les modulateurs Sigma Delta passe bande : notre application	48
III.3.2.1.	Les modulateurs Sigma Delta passe bande réels	49
III.3.2.2.	Les modulateurs Sigma Delta passe bande complexes	51
IV.	LES FILTRES A CAPACITES COMMUTEES	53
IV.1.	Rappel théorique et exemples	54
IV.1.1.	Bases des filtres à capacités commutées	54
IV.1.2.	Structures avec rétroaction capacitive	55
IV.1.3.	Structures avec rétroaction non capacitive	56
IV.2.	Mise en œuvre des circuits à capacités commutées	57
IV.2.1.	La non-linéarité de la résistance R_{ON} des interrupteurs MOS	58
IV.2.2.	Les injections d'horloge des interrupteurs MOS	58
IV.2.3.	Les capacités MOS	59
IV.3.	Circuits générateurs d'horloges	60
V.	CONCLUSION	61
CHAPITRE II		63
I.	INTRODUCTION	65
II.	ETUDE THEORIQUE DES BRUITS	67
II.1.	Rappels	67
II.1.1.	Calcul de la puissance	67
II.1.2.	Unités de mesure de la Densité Spectrale de Puissance	68
II.1.2.1.	Le dB_c / Hz	69
II.1.2.2.	Le $dB_{m50\Omega} / Hz$	70
II.1.3.	Analyse spectrale	71
II.2.	Le bruit au niveau des composants électroniques	72
II.2.1.	Le bruit thermique	72
II.2.2.	Le bruit de grenaille	73

II.2.3.	Le bruit de Flicker	73
II.3.	Manifestation du bruit dans les circuits à temps discret	74
II.3.1.	Bruit blanc de tension	74
II.3.2.	Bruit d'amplitude d'une sinusoïde	75
II.3.3.	Bruit blanc d'échantillonnage	77
II.3.4.	Généralisation	79
II.4.	Bruit de phase d'un oscillateur	81
II.4.1.	Origine du bruit de phase	82
II.4.2.	Spectre du bruit de phase d'une sinusoïde	83
II.4.3.	Influence du jitter FM sur l'échantillonnage d'une porteuse	84
III.	MODELISATION DES BRUITS	86
III.1.	Le générateur de gaussienne	86
III.1.1.	Principe	86
III.1.2.	Modèle	86
III.1.3.	Performances	87
III.2.	Le bruit blanc de tension	88
III.2.1.	Principe	88
III.2.2.	Modèle	88
III.2.3.	Performances	89
III.3.	Le bruit de tension en 1/f	90
III.3.1.	Principe	90
III.3.2.	Modèle	92
III.3.3.	Performances	93
III.4.	Le bruit de phase	94
III.4.1.	Principe	94
III.4.2.	Modèle	95
III.4.3.	Performances	95
IV.	MODELISATION DES GENERATEURS DE SIGNAUX	98
IV.1.	Le générateur de sinusoïdes	98
IV.1.1.	Principe	98
IV.1.2.	Modèle	99
IV.1.3.	Performances	100

IV.2. Le générateur d'horloge	102
IV.2.1. Principe	102
IV.2.2. Modèle	103
IV.2.3. Performances	104

V. CONCLUSION	107
----------------------	------------

CHAPITRE III	109
---------------------	------------

I. INTRODUCTION	111
------------------------	------------

II. LOCALISATION DES EFFETS DU JITTER INTERNE	112
--	------------

III. PRESENTATION DU MONTAGE DE TEST ET DU PROTOCOLE DE MESURE	115
---	------------

III.1. Préambule	115
------------------	-----

III.2. La structure de test	116
-----------------------------	-----

III.3. La méthode de mesure	118
-----------------------------	-----

III.3.1. Bilan des bruits	118
---------------------------	-----

III.3.2. L'histogramme verrouillé	121
-----------------------------------	-----

IV. MESURES ET INTERPRETATIONS	125
---------------------------------------	------------

IV.1.1. Présentation du banc de mesure	125
--	-----

IV.1.2. Mesure type	127
---------------------	-----

IV.1.3. Généralisation des mesures	129
------------------------------------	-----

V. CONCLUSION	130
----------------------	------------

CHAPITRE IV	131
--------------------	------------

I. INTRODUCTION	133
------------------------	------------

II. PREAMBULE	135
----------------------	------------

III. MODELISATION DE LA CHAINE FI (FREQUENCE INTERMEDIAIRE)	138
--	------------

III.1. L'amplificateur à Contrôle Automatique de Gain (CAG)	138
III.1.1. Etude théorique du circuit	138
III.1.2. Simulations mixtes (modèle transistor / VHDL-AMS)	138
III.2. Le générateur de quadrature	140
III.2.1. Etude théorique du circuit	140
III.2.2. Simulations mixtes (modèle transistor / VHDL-AMS)	141
III.3. Le mélangeur	142
III.3.1. Etude théorique du circuit	142
III.3.2. Simulations mixtes (modèle transistor / VHDL-AMS)	143
III.4. Le filtre anti-repliements complexe (Anti-Aliasing Filter)	144
III.4.1. Etude théorique du circuit	144
III.4.2. Simulations mixtes (modèle transistor / VHDL-AMS)	145
III.5. La boucle à verrouillage de phase (PLL)	146
III.5.1. Etude théorique du circuit	146
III.5.2. Simulations mixtes (modèle transistor / VHDL-AMS)	147
III.6. La chaîne FI	147
IV. MODELISATION DU MODULATEUR SIGMA DELTA	150
IV.1. Le générateur d'horloges non-recouvrantes	150
IV.1.1. Etude théorique du circuit	150
IV.1.2. Simulations mixtes (modèle transistor / VHDL-AMS)	150
IV.2. Le Convertisseur Analogique/Numérique	151
IV.2.1. Etude théorique du circuit	151
IV.2.2. Simulations mixtes (modèle transistor / VHDL-AMS)	152
IV.3. Le Convertisseur Numérique/Analogique	153
IV.3.1. Etude théorique du circuit	153
IV.3.2. Simulations mixtes (modèle transistor / VHDL-AMS)	154
IV.4. Le résonateur	155
IV.4.1. Etude théorique du circuit	155
IV.4.2. Simulations VHDL-AMS	155
IV.5. Le modulateur Sigma Delta	156
IV.5.1. Etude théorique du circuit	156
IV.5.2. Simulations VHDL-AMS	157

V.	MODELISATION DE LA CHAINE DE RECEPTION RADIO-FREQUENCE	158
V.1.	Le mesure du SNR	158
V.2.	Les contributions en bruit	160
V.2.1.	Le jitter	160
V.2.2.	Le bruit de phase	162
V.2.2.1.	Les sources	162
V.2.2.2.	Le modèle théorique	163
VI.	CONCLUSION	166
	CONCLUSION GENERALE	167
	BIBLIOGRAPHIES	171
	ANNEXES	179

INTRODUCTION

Il y a plus de 100 ans le mot "Télécommunications" était créé par Édouard Estaunié. Pourtant ce n'est que depuis les vingt dernières années, avec l'explosion du marché des radiocommunications de type GSM et plus récemment de l'internet, que ce terme est pleinement entré dans le langage courant. Après être restée un temps l'apanage des professionnels et des militaires, la radiocommunication a vu son marché augmenter exponentiellement avec l'ouverture au grand public. Cet accroissement du marché s'est accompagné d'un transfert d'une partie des technologies employées dans la radiocommunication professionnelle vers les technologies grand public. Les technologies spécifiques à la Radiocommunication Mobile Professionnelle ou PMR (Private Mobile Radio) ont alors été dépassées, et ont failli disparaître. Cependant les besoins de ce type de téléphonie en terme de coûts et de performances (confidentialité, débit de données, qualité d'écoute et de puissance) n'étant pas ceux du grand public, la totalité des composants grand public n'ont pu y être employés.

La tendance actuelle des technologies grand public est une diminution de la part de l'électronique analogique dans le traitement des signaux. La part croissante des technologies numériques s'explique, entre autres, par une plus grande flexibilité, une meilleure reproductibilité des performances et une rentabilité accrue de ces technologies par rapport aux technologies analogiques. C'est le principe de la radio logicielle qui tend à numériser les signaux au plus proche de l'antenne d'émission/réception. Cependant malgré d'énormes progrès réalisés ces dernières années dans le domaine de la conversion analogique/numérique, il n'est toujours pas possible à l'heure actuelle de se passer complètement d'une mise en forme analogique des signaux. Dans le cas d'une chaîne de réception, certaines spécifications de la mise en forme analogique ont pu être allégées par le traitement numérique, c'est le cas notamment de la sélection numérique du canal qui a entraîné une réduction de la complexité des filtres analogiques. Parallèlement, la réduction de la place occupée par le bloc analogique a engendré une augmentation des exigences en terme de performance du bloc de conversion analogique/numérique.

Pour faire face à la spécificité de leurs besoins et à la concurrence des appareils GSM, l'industrie de la PMR a alors réagi en migrant, à son tour, vers les technologies numériques afin de développer ses propres solutions et ainsi reprendre sa place de précurseur.

Les travaux de thèse que nous présentons concernent l'évaluation des performances en bruit du bloc analogique et mixte de la chaîne de réception radiofréquence et plus particulièrement du convertisseur analogique/numérique situé en fin de chaîne. Le but de ces travaux n'est pas d'apporter de nouvelles solutions pour réduire les sources de bruit, mais plutôt de fournir aux concepteurs de circuits radiofréquences les modèles comportementaux nécessaires à l'évaluation de l'impact des sources de bruit sur certains types d'architecture. Pour vérifier nos théories, nos modèles ainsi que les paramètres physiques de ces modèles, nous disposerons d'un circuit démonstrateur développé par le groupe ASIC (Application Specific Integrated Circuit) de la société EADS-TELECOM (société spécialisée dans les télécommunications professionnelles). Cet ASIC mixte (circuit qui intègre des fonctions analogiques et numériques) a été caractérisé et comporte une chaîne de réception radiofréquence complète. Il est destiné aux portatifs ainsi qu'aux stations de base.

Le document suivant se décompose en quatre chapitres qui reprennent les deux grands axes de travail de la thèse : la modélisation comportementale des sources de bruit des blocs analogiques d'une part et la caractérisation des sources de bruit d'autre part.

Le premier chapitre pose les bases techniques nécessaires à la compréhension du fonctionnement des divers blocs constituant la chaîne de réception radiofréquence. Ce chapitre part d'un rapide survol des diverses architectures de chaîne réception RF pour expliquer le choix effectué par la société. Une fois cette architecture détaillée, notre intérêt se porte sur les structures de conversion analogique/numérique. Après les rappels d'usage sur ce type de conversion, cette partie se consacre aux modulateurs Sigma Delta et plus particulièrement aux architectures passe bande à temps discret. La dernière partie de ce chapitre est dédiée aux filtres à capacités commutées utilisés dans ce type de modulateur. Cette étude des blocs constituant la chaîne nous permettra de localiser les différentes sources de bruit.

Le second chapitre est consacré à l'étude et à la caractérisation des sources de bruits. Ce chapitre est le cœur de la thèse. Il débute par une étude approfondie des diverses sources de bruits et permet d'extraire les relations nécessaires à leur modélisation. Ces relations sont valables quel que soit le simulateur, et peuvent ainsi être implémentées dans de nombreux langages de modélisation. La seconde partie du chapitre est justement consacrée à la modélisation de ces bruits en langage VHDL-AMS. Nous développons dans cette partie une méthodologie permettant de modéliser la plupart des formes de bruit. La dernière partie développe l'intégration de ces bruits dans des générateurs de signaux (sinusoïdaux et carrés). A l'issue de ce chapitre, nous disposerons d'une bibliothèque de modèles VHDL-AMS permettant la prise en compte du bruit au niveau comportemental.

Le troisième chapitre permet de répondre à la problématique posée en fin du second chapitre à savoir qu'un modèle ne peut reproduire la réalité sans paramètre physique réaliste. Pour cela, la valeur du jitter interne d'une structure simple à capacités commutées est évaluée. Ce chapitre est consacré à la mise en œuvre et à l'interprétation de cette mesure. La première partie justifie l'utilisation d'une structure moins complexe qu'un modulateur Sigma Delta ainsi que la démarche suivie. La seconde décrit la structure de test et le protocole de mesure. Les résultats de mesure ainsi que leur interprétation sont reportés dans la dernière partie du chapitre.

Nous faisons converger, dans le quatrième chapitre, les études et les modélisations réalisées dans les précédents chapitres. Le but de ce chapitre est de reproduire a posteriori, en simulation, les performances (et les contre-performances) de la chaîne de réception RF développée par la société EADS-TELECOM. Cette étape est nécessaire pour valider les modèles proposés avant d'en dégager une démarche d'évaluation prédictive pour les circuits à venir. Pour cela, nous réalisons dans la première partie une modélisation bloc à bloc de la chaîne FI (Fréquence Intermédiaire) située avec le modulateur Sigma Delta. Chaque modèle en VHDL-AMS intègre les principales caractéristiques électriques du bloc et notamment le bruit interne. La seconde partie concerne la modélisation en VHDL-AMS du modulateur Sigma Delta. A l'issue de ces deux parties, l'ensemble de la chaîne de réception RF est modélisé. La troisième partie est consacrée à la simulation de la chaîne analogique complète (associant la chaîne FI et le modulateur Sigma Delta) et la comparaison des simulations avec les mesures.

La conclusion générale sera l'occasion de tirer un bilan de ces travaux réalisés durant cette thèse ainsi que d'esquisser les perspectives qui pourront être données à ce travail.

CHAPITRE I :

Les Convertisseurs Analogique/Numérique
Sigma Delta à temps discret dans
les chaînes de réception Radio-Fréquence

I. INTRODUCTION

Nous avons défini dans l'introduction générale le contexte économique et scientifique de la thèse. Il nous faut maintenant la positionner dans son contexte technique, c'est-à-dire définir les bases techniques à partir desquelles nous allons réaliser nos travaux. La mise en place de ces définitions techniques est la raison de chapitre.

Ce chapitre est construit selon une approche « Macro/Microscopique », soit de la vue la plus large vers la structure la plus précise. Le but de la thèse étant l'étude, la caractérisation et la modélisation des bruits dans les structures à capacités commutées que sont les modulateurs Sigma Delta temps discret utilisés dans une chaîne de réception Radio Fréquence (RF), nous débuterons tout naturellement ce chapitre par un rapide survol des différentes architectures de réception RF. Une fois cette description effectuée, nous présenterons l'architecture retenue. L'étude de son cahier des charges nous permettra de définir les principales caractéristiques du convertisseur analogique/numérique. Après un rapide rappel sur les principes de bases qui régissent la conversion analogique/numérique, nous concentrerons notre présentation, dans la seconde partie, sur la modulation Sigma Delta. Nous en profiterons pour décrire quelques architectures passe bas, pour ensuite nous attacher à la description du modulateur utilisé dans la chaîne de réception RF, soit le modulateur à temps discret passe bande réel ou complexe. La troisième partie sera axée sur l'étude des filtres à capacités commutées utilisés par les modulateurs Sigma Delta.

A l'issue de ce chapitre nous serons pleinement capables d'appréhender l'environnement technique de notre étude et par là même de débiter les travaux d'investigation.

II. Les systèmes de radiocommunication

Qu'est-ce qu'un système de radiocommunication ? C'est ce que nous allons définir dans un premier temps, pour ensuite nous concentrer sur la partie réception de l'ensemble. Nous terminerons par les évolutions passées et futures dont découlent les contraintes actuelles.

II.1. Chaîne complète d'émission/réception

Dans le cas idéal, le système Radio Fréquence (RF) pourrait se représenter ainsi :

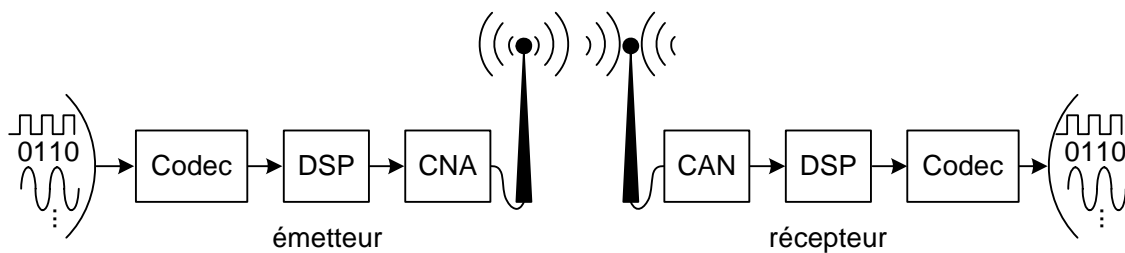


Figure 1 - 1 : Chaîne d'émission – réception « radio logicielle »

En début de chaîne, les informations envoyées par l'émetteur seraient codées numériquement par un codeur décodeur (Codec), puis modulées et transposées sur la fréquence porteuse dans le DSP (Digital Signal Processor) ou dans tout autre type de système de traitement du signal. Pour finir, elles seraient converties en signaux analogiques par un CNA (Convertisseur Analogique/Numérique) pour être émises directement par l'antenne. La chaîne de réception effectuerait le cheminement inverse. Les signaux analogiques seraient convertis en signaux numériques par un CAN (Convertisseur Numérique/Analogique) dès la sortie de l'antenne, puis les opérations de sélection de canal, de démodulation et de décodage s'effectueraient dans le DSP.

C'est le concept du « software radio » ou « radio logicielle » apparu au sein de la recherche militaire à la fin des années 70 et élargi aux télécommunications civiles dans les années 90 [Geg02].

Même si la tendance actuelle vise ce type de réalisations, il est encore technologiquement impossible de les implanter. La principale limitation provient des CAN dont la dynamique et le rapport fréquence de travail - puissance dissipée sont encore insuffisants pour que l'on puisse s'affranchir de l'utilisation d'amplificateurs faible bruit (Low Noise Amplifier ou LNA) placés en début de chaîne. Viennent s'ajouter les capacités de traitement insuffisantes des processeurs et une trop grande consommation d'énergie due aux nombreux opérations et calculs. Les chaînes d'émission - réception ne peuvent donc pas encore se passer de la mise en forme analogique du signal (Figure 1 - 2).

Outre une amplification sélective, cette mise en forme comprend également le filtrage, la transposition en basse fréquence des signaux et parfois la sélection de canal.

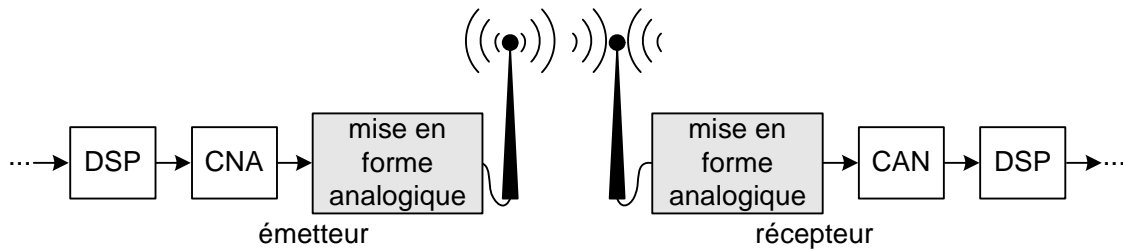


Figure 1 - 2 : Principe actuel d'une chaîne d'émission – réception

La suite de ce chapitre sera uniquement consacrée à la mise en forme analogique et à la conversion analogique/numérique de la partie réception de la chaîne.

II.2. Les architectures de réception

Les récepteurs radio fréquences sont caractérisés par leur sensibilité et leur sélectivité. La sensibilité définit la capacité du système à détecter et à amplifier le signal recherché quand son amplitude est faible. La sélectivité évalue la capacité du système à détecter et à amplifier le signal recherché lorsque celui ci est entouré de signaux parasites d'amplitude plus forte.

Il existe plusieurs types d'architecture de récepteur dont les plus connus sont les architectures hétérodynes [Lee98], homodynes ou les architectures à faible fréquence intermédiaire. Quelle que soit l'architecture, le principe de réception est une variante de celui décrit ci-dessous :

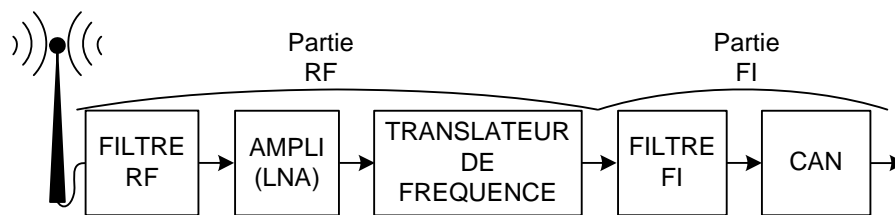


Figure 1 - 3 : Principe de mise en forme analogique d'un récepteur RF

La première étape consiste à isoler et amplifier le canal RF recherché à l'aide d'un filtre RF passe bande et d'un amplificateur faible bruit. L'opération de sélection analogique de canal lorsque la largeur de canal est faible, nécessite des filtres très sélectifs difficilement réalisables lorsque la fréquence de travail est élevée. De même la conversion analogique/numérique réclame une faible fréquence de travail pour la dynamique requise.

Il est alors nécessaire de traduire la fréquence de travail à une fréquence intermédiaire fixe f_i plus basse que celle de la partie RF par l'usage d'un mélangeur qui réalise une multiplication entre le

signal à tradater S_u et un signal issu d'un oscillateur local S_{OL} . Nous notons f_{OL} la fréquence de l'oscillateur local et f_u la fréquence autour de laquelle est centrée le signal utile avant sa translation à la fréquence intermédiaire $f_i = f_u - f_{OL}$:

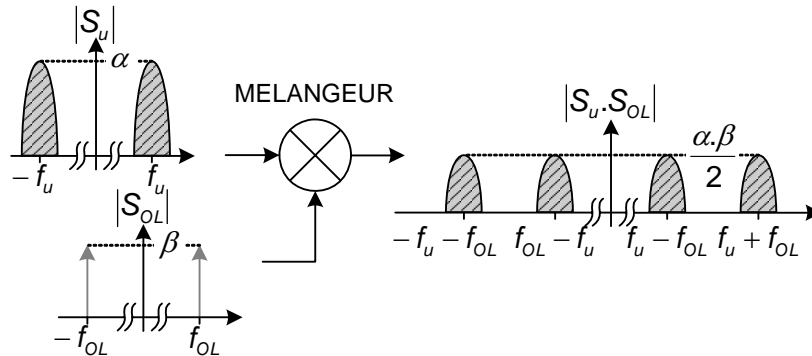


Figure 1 - 4 : Principe du mélangeur

N.B. : Lorsque $f_{OL} > f_u$ l'architecture est dite supradynne et inversement infradyne lorsque $f_u > f_{OL}$.

Nous nous intéressons maintenant aux trois types d'architecture de récepteur énoncés précédemment.

II.2.1. L'architecture hétérodyne

II.2.1.1. *Le concept*

Cette architecture a été conçue par Fessenden en 1910 [Lar94]. Nous décrivons ci-dessous une première version simplifiée (cf. Figure 1 - 5) composée d'une antenne, d'un premier filtre passe bande placé juste après l'antenne qui permet de sélectionner la bande RF, c'est généralement un filtre à onde de surface (SAW : Surface Acoustic Wave), d'un amplificateur faible bruit (LNA), d'un mélangeur couplé avec un oscillateur local (OL) et d'un filtre de sélection de canal qui élimine les composantes hautes fréquences $(-f_u - f_{OL}; f_u + f_{OL})$.

Cette architecture est la plus basique par son concept mais également la moins performante. En effet considérons, comme décrit ci-dessous, un signal utile centré autour de f_u et un signal perturbateur centré autour de f_p , également appelée fréquence image, tels que les deux signaux soient séparés par une bande de fréquence égale à $2.f_i$:

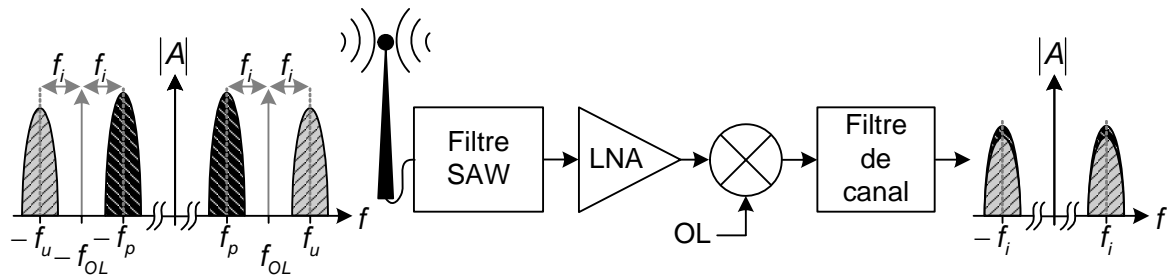


Figure 1 - 5 : Principe du récepteur hétérodyne et illustration de l'effet de la fréquence image

L'utilisation du mélangeur induit un repliement du signal perturbateur sur le signal utile, ce qui n'est pas acceptable. Il faut donc supprimer ou du moins très fortement diminuer la puissance du perturbateur avant son entrée dans le mélangeur. On utilise pour cela un filtre de réjection d'image.

N.B. : La figure ci-dessous illustre le cas où $(f_u = f_{OL} + f_i; f_p = f_{OL} - f_i)$, nous aurions pu également traiter le cas où $(f_u = f_{OL} - f_i; f_p = f_{OL} + f_i)$.

II.2.1.2. La réjection d'image

Comme décrit par la Figure 1 - 6, un filtre est introduit entre l'amplificateur faible bruit et le mélangeur. Ce filtre dit « filtre à réjection d'image » permet d'atténuer le signal image centré sur f_p .

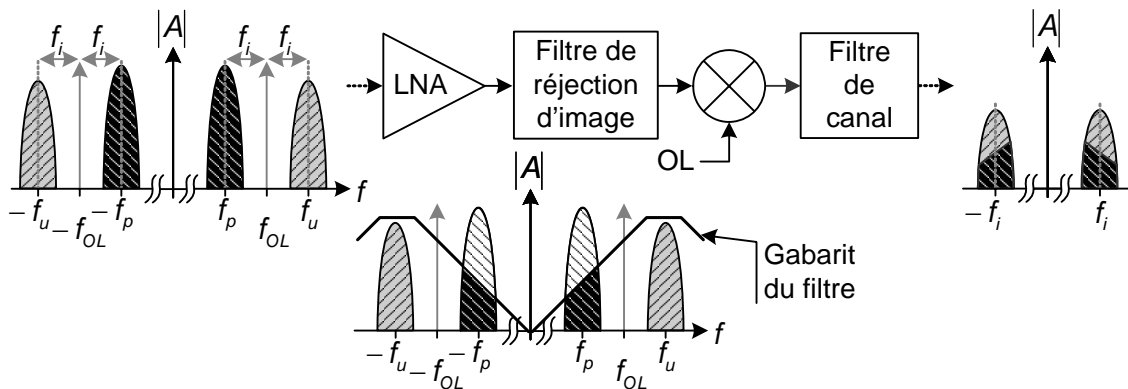


Figure 1 - 6 : Filtre à réjection d'image

Cette solution n'est cependant pas pleinement satisfaisante. En effet, l'efficacité du filtre à réjection d'image décroît lorsque les fréquences f_u et f_p sont proches c'est-à-dire lorsque f_i est faible. A l'inverse, le filtre de canal a une efficacité accrue pour de faibles f_i . L'utilisation d'un double changement de fréquence permet de concilier ces deux contraintes. Une première fréquence intermédiaire élevée permet d'optimiser le fonctionnement du filtre à réjection d'image. Une seconde fréquence intermédiaire plus basse permet ensuite d'augmenter la sélectivité du filtre de canal.

II.2.1.3. L'architecture hétérodyne à double changement de fréquence

La figure ci-dessous nous fournit un exemple d'une architecture à double changement de fréquence :

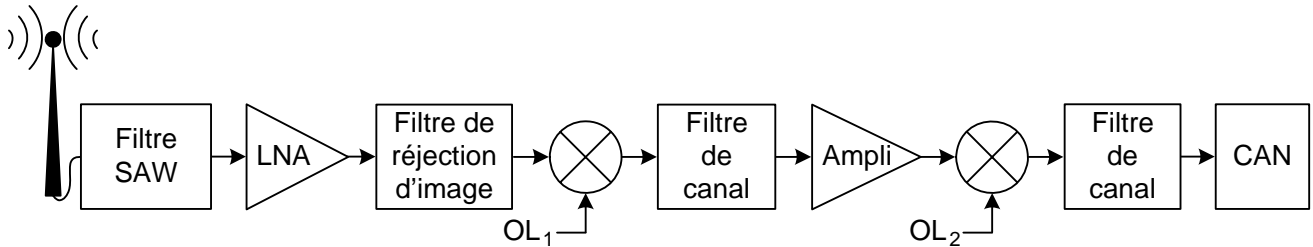


Figure 1 - 7 : Architecture hétérodyne à double changement de fréquence

Le fonctionnement du début de la chaîne est similaire à celui énoncé précédemment. Afin de conserver l'amplitude du signal un amplificateur est placé juste après le filtre de canal. Cet amplificateur a la particularité d'être sélectif dans une certaine gamme de fréquences. Après cette amplification, un second mélangeur abaisse de nouveau la fréquence de travail. Après un second filtrage, le signal est envoyé sur le convertisseur analogique/numérique. Le second filtre de canal peut être remplacé par un filtre anti-repliement si la conversion analogique/numérique ne s'effectue pas en bande de base. En effet la conversion analogique/numérique nécessite une discrétisation du signal, ce qui implique un échantillonnage et donc une périodisation du spectre avec un éventuel recouvrement. Ce filtre a alors pour rôle d'atténuer (cf. Figure 1 - 55) les composantes fréquentielles du signal au-delà de $F_{ech}/2$.

II.2.1.4. L'architecture hétérodyne polyphasée

Ce type d'architecture est préféré pour des systèmes à modulation numérique. En effet, contrairement à la modulation d'amplitude, les modulations de fréquence ou de phase peuvent induire des spectres non symétriques par rapport au DC. Il faut donc pouvoir différencier les composantes aux fréquences positives des composantes aux fréquences négatives.

Le principe est celui de l'architecture hétérodyne jusqu'à l'amplification. La différence vient du mélangeur où le signal est dédoublé en deux signaux en quadrature de phase. Cette opération fournit, après filtrage, deux signaux I et Q (In phase and Quadrature) déphasés de $\pi/2$. La Figure 1 - 8 donne un exemple d'une telle architecture.

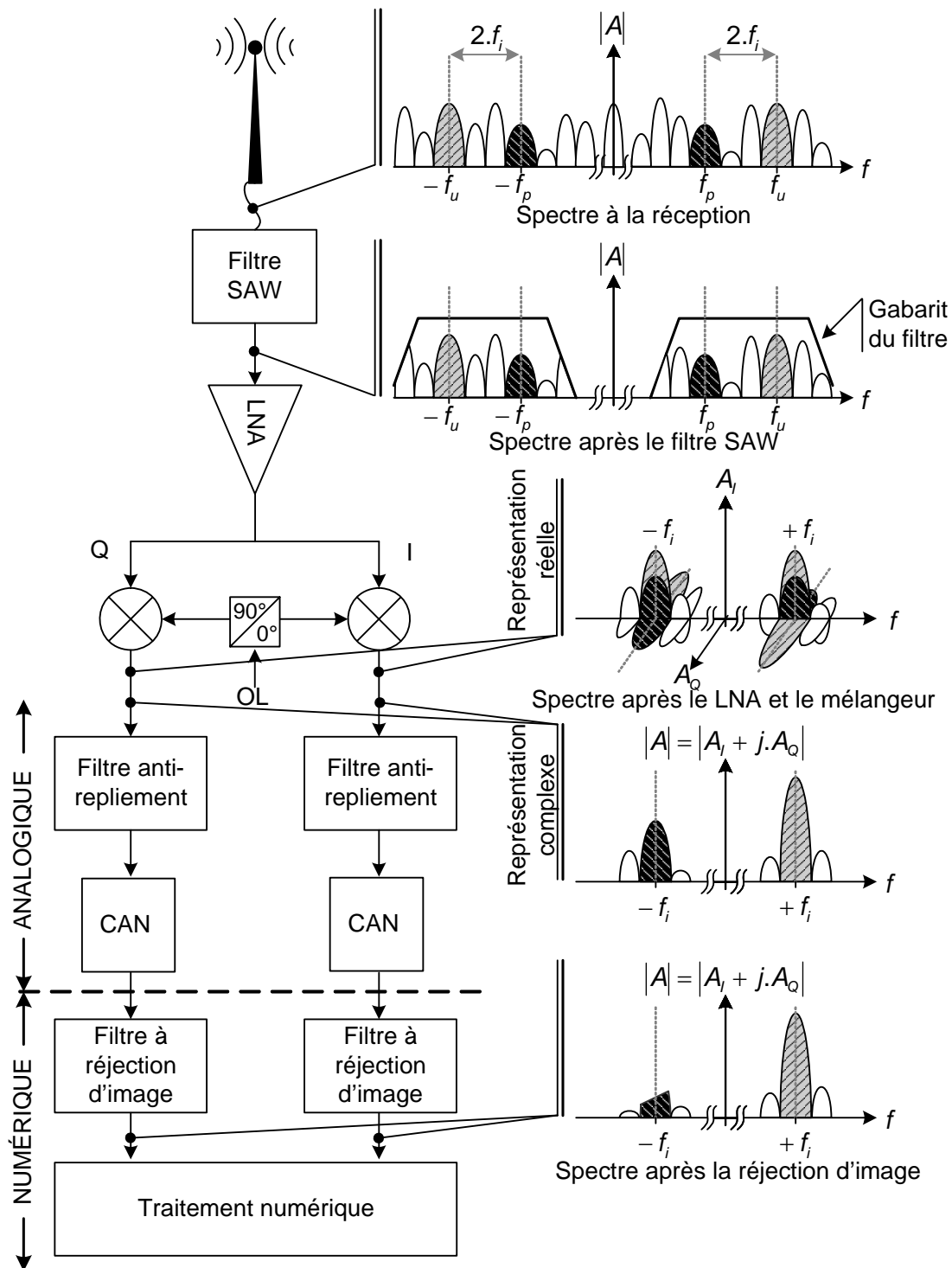


Figure 1 - 8 : Architecture hétérodyne polyphasée

N.B. : Comme dans l'architecture présentée ci-dessus, le filtre à réjection d'image peut être placé après la conversion analogique/numérique. La réjection d'image est alors réalisée par des filtres numériques, ce qui permet une plus grande souplesse d'utilisation. Cette technique de réjection d'image utilisant des filtres polyphasés a été introduite par J. C. Rudell [Rud97].

Deux représentations spectrales sont possibles, soit les voies I et Q sont considérées comme indépendantes soit comme composantes d'un même signal (l'illustration est également disponible Figure 1 - 10). Les bonnes performances en terme de sélectivité et de sensibilité expliquent en grande partie le succès de ce type d'architecture dans les mobiles GSM de deuxième génération. Cependant un prix et une consommation élevés ainsi que le besoin de filtres externes de haute qualité, réduisant son intégration, restent des facteurs limitants.

N.B. : Le terme d'architecture superhétérodyne est employé lorsque le premier oscillateur local n'est pas à fréquence fixe et permet ainsi un balayage de plusieurs canaux. Souvent, l'invention de ce concept est attribuée, à tort, à l'américain Edwin Howard Armstrong en 1917 [Lar95], alors que c'est un français Lucien Lévy [Hap66] [Bre17] qui, le premier, a déposé un brevet décrivant ce type d'architecture.

II.2.2. L'architecture homodyne

II.2.2.1. Le concept

Ces architectures, également désignées par les termes « zero-IF » ou « à conversion directe » [Abi95] [Raz97], ont un fonctionnement proche de celui précédemment énoncé, à savoir l'abaissement de la fréquence de travail à une fréquence plus faible pour rendre possible son traitement. La principale différence vient de la fréquence de l'oscillateur local qui est alors égale à la fréquence du signal utile. La fréquence intermédiaire est alors nulle et la sortie du système se situe directement en bande de base, ce qui évite ainsi l'utilisation d'un filtre de réjection d'image.

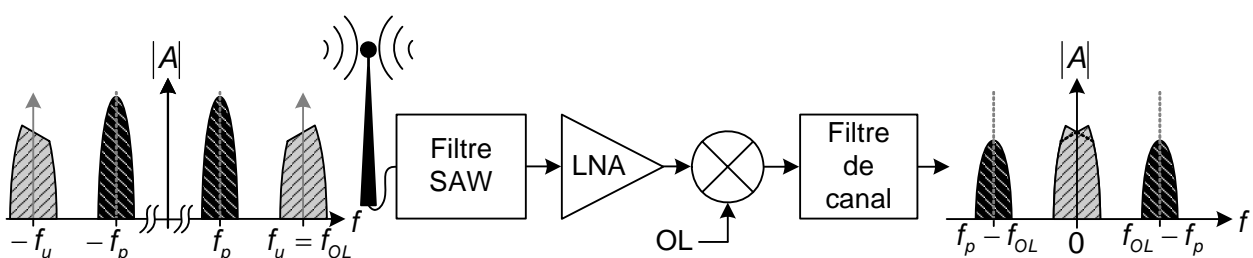


Figure 1 - 9 : Principe du récepteur homodyne

Ce type d'architecture n'est pas adapté pour des spectres à bandes latérales non symétriques. Comme illustré par la figure ci-dessus, le décalage en fréquence a engendré le repliement du spectre du signal utile sur lui-même, ce qui se traduit par la dégradation de l'information. Pour y remédier des structures polyphasées sont alors utilisées.

II.2.2.2. L'architecture homodyne polyphasée

Comme précédemment (cf. §II.2.1.4) l'utilisation d'un mélangeur quadratique permet de différencier les parties positives et négatives du spectre. La figure suivante illustre le fonctionnement d'un tel mélangeur :

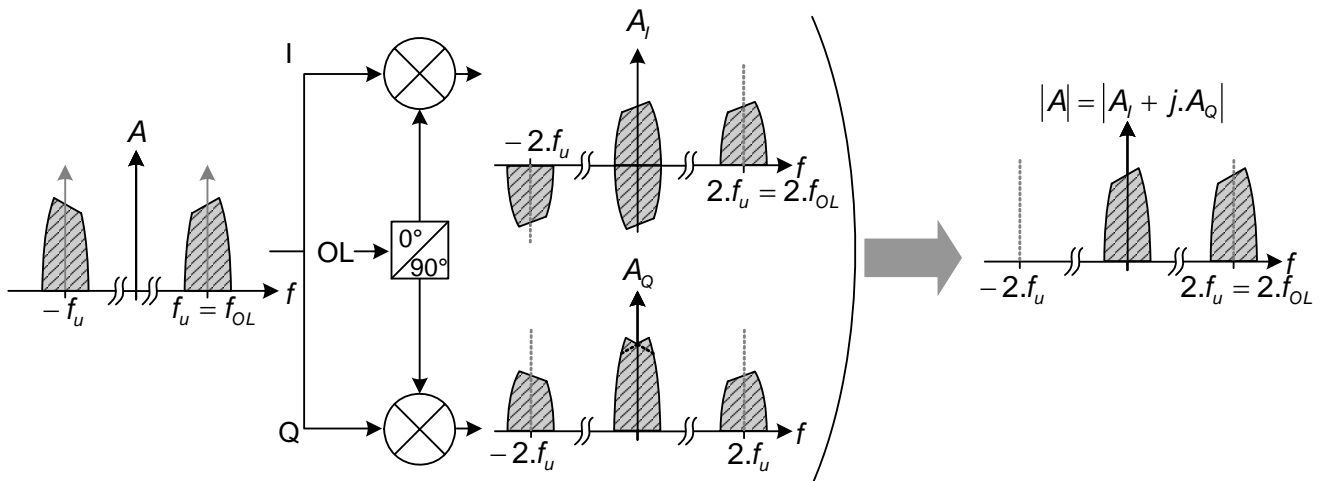


Figure 1 - 10 : Mélangeur quadratique

L'architecture homodyne est intéressante du fait de sa simplicité, de l'absence de filtre de réjection d'image et d'une faible consommation. De plus elle est fortement intégrable. Cependant cette structure est très sensible à l'apparition de composantes continues qui peuvent venir corrompre le signal, ainsi qu'aux bruits basses fréquences, comme le bruit de Flicker, qui ne sont plus négligeables en bande de base.

II.2.3. L'architecture à faible fréquence intermédiaire

L'architecture à faible fréquence intermédiaire (ou low IF) peut être considérée comme une synthèse des deux précédentes architectures. C'est en fait un montage hétérodyne avec une sortie proche de la bande de base (de quelques kHz à quelques MHz), ce qui réduit sa sensibilité aux bruits des basses fréquences et qui simplifie le filtrage en F.I. Comme énoncé précédemment, l'architecture low IF peut être à simple ou double changement de fréquence et/ou polyphasée. De plus, le filtre de réjection d'image peut être remplacé par un filtre polyphasé comme ceux développés par Hartley [Har28] ou Weaver [Wea56]. Même si les évolutions futures semblent prometteuses, ce type d'architecture est encore trop soumis aux perturbations liées à l'intermodulation d'ordre 2, aux fuites de l'oscillateur local dans le circuit ainsi qu'au désappariement de gain et de phase qui entraînent des dissymétries entre les voies I et Q réduisant ainsi la réjection d'image.

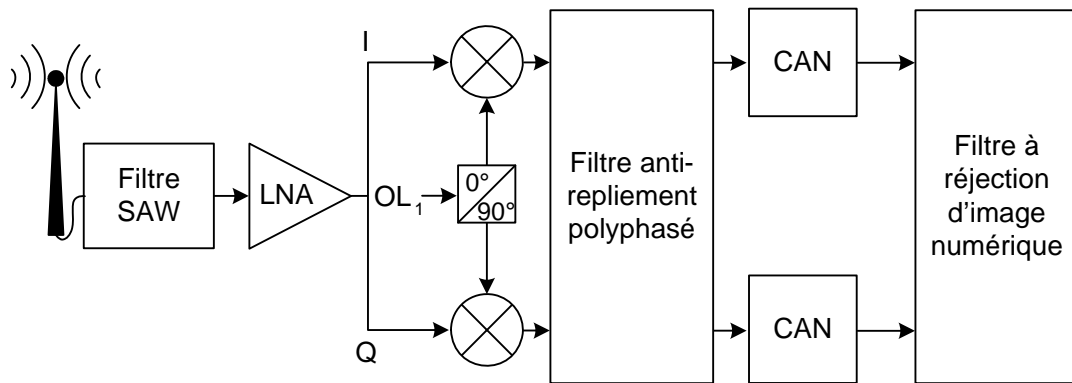


Figure 1 - 11 : Principe du récepteur low IF

II.3. L'architecture de réception retenue

Le récepteur conçu par le service ASIC de la société EADS – TELCOM entre dans la composition des PMR (Private or Professional Mobil Radiocommunication). Son cahier des charges est le suivant : forte intégration, faible coût, faible encombrement. Le récepteur doit également être multi norme.

L'architecture retenue par la société EADS - TELECOM est fondée sur une architecture low IF polyphasée. Comme cette architecture reste assez contraignante techniquement, des modifications ont été apportées.

La sélection analogique de canal nécessite un oscillateur ayant un pas de synthèse fin, un gabarit pour le filtre de canal très étroit et un amplificateur entre le premier et le second mélangeur très sélectif. Cela entraîne la fabrication de synthétiseurs de fréquences complexes, de filtres très sélectifs d'ordres élevés (supérieur à 4). Leur réalisation met alors en œuvre des moyens technologiques lourds et coûteux qui nécessitent un grand nombre de composants discrets, gourmands en place et en consommation.

Or comme cela a été expliqué dans l'introduction, la tendance actuelle pousse au contraire à une très large intégration de l'ensemble des architectures ainsi qu'à la mixité des circuits (les parties analogiques et numériques cohabitent sur une même puce). Ce type de structure n'est donc pas adapté, il faut alors revoir la philosophie du circuit.

Comme la sélection analogique du canal est dispendieuse en composants discrets, en coût et en consommation, le choix a été fait de la réaliser numériquement. Cette alternative a été rendue possible par une forte progression de la densité d'intégration des fonctions logiques

(≈ 14 kportes/mm² en technologie 0.35 μ m à ≈ 62 kportes/mm² en technologie 0.18 μ m) qui permet la synthèse de filtres numériques d'ordres élevés.

Le déplacement de la sélection de canal a permis de « relâcher » les contraintes sur le premier oscillateur et sur le filtre de sélection de canal. Les signaux à convertir étant moins « propres », moins filtrés et ayant un spectre en fréquence plus étendu, les contraintes, notamment de linéarité et de bruit, se sont alors reportées en partie sur les CAN.

L'architecture retenue est la suivante :

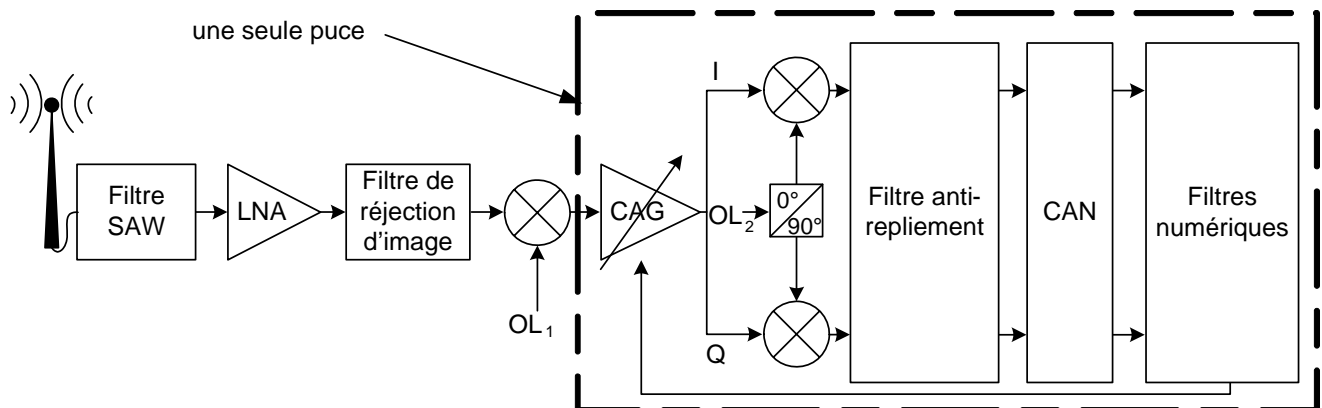


Figure 1 - 12 : Architecture du récepteur

La puce étudiée comporte une grande partie de la mise en forme analogique. L'autre partie est réalisée en éléments discrets.

A l'entrée de la puce la fréquence de porteuse est invariablement de 73.125MHz. Un amplificateur à Contrôle Automatique de Gain (CAG) a été préféré à un amplificateur classique. Cet amplificateur permet, outre une amplification sélective, de maintenir constante la puissance du signal entrant afin d'utiliser la chaîne de manière optimale (pas d'effet de saturation). Le contrôle du gain s'effectue par rebouclage à la sortie des filtres numériques. La fréquence du second oscillateur est de 71.5MHz. Les signaux en quadrature de phase sont donc à la fréquence 1.625MHz. La conversion analogique/numérique s'effectue, elle, à 13MHz, soit 8 fois la fréquence du signal d'entrée.

Dans la suite, nous nous intéresserons au fonctionnement des convertisseurs analogiques numériques, ainsi qu'aux architectures de conversion mises en place pour répondre aux contraintes de dynamique et de bruit.

III. La conversion analogique/numérique utilisant la modulation Sigma Delta

III.1. La conversion analogique/numérique « classique » : généralités

La conversion de données de l'analogique vers le numérique permet de passer d'un signal continu dans le temps et de valeur continue à un signal temps discret et à valeur discrète [DeQ96] :

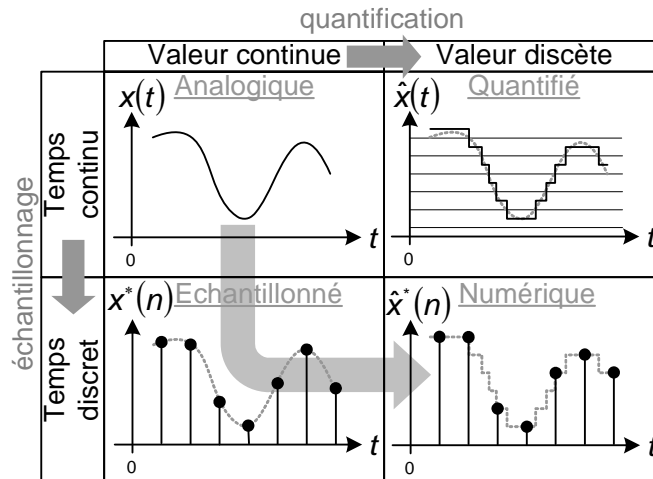


Figure 1 - 13 : Les quatre états d'un signal

La discrétisation, à l'aide d'un quantificateur, des valeurs continues provoque une perte d'informations. Cette perte d'informations ou erreur de quantification induit le bruit de quantification. Dans un monde idéal, la quantification serait instantanée. Ce point est techniquement impossible pour le moment. Il faut donc prélever à période fixe la valeur du signal à convertir (échantillonner) et maintenir la valeur constante (bloquer) le temps nécessaire à la quantification. C'est le rôle de l'échantillonneur bloqueur. L'étude de l'effet de ces opérateurs, d'abord séparément puis assemblés, sur un signal à temps continu sera l'objet de cette première sous-partie.

III.1.1. L'échantillonneur bloqueur

D'un point de vu fonctionnel, l'échantillonneur bloqueur est au minimum composé (cf. Figure 1 - 14) d'un interrupteur qui joue le rôle d'échantillonneur et d'une capacité qui joue le rôle de bloqueur en mémorisant la valeur échantillonnée le temps nécessaire à sa quantification ($T_{ech} - \tau$ dans notre exemple). Comme l'échantillonneur n'est pas idéal, le spectre à l'issue de l'échantillonnage est égal au produit de convolution des spectres des signaux $x(t)$ et $r(t)$ (cf. §I.1.2).

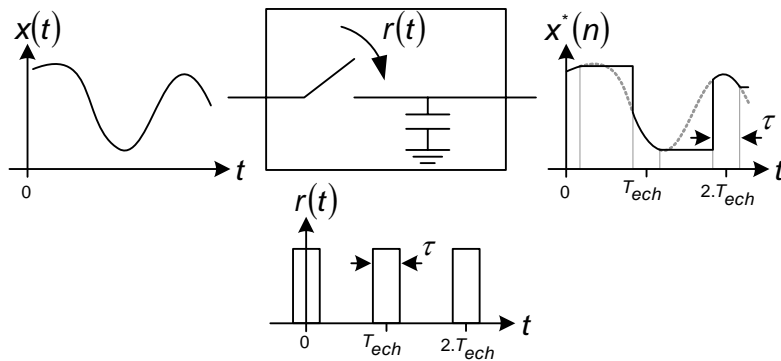


Figure 1 - 14 : Représentation temporelle du fonctionnement d'un échantillonneur bloqueur

Pour comprendre l'effet du bloqueur sur le spectre du signal issu de l'échantillonnage réel, il faut nous placer du point de vue du quantificateur. En effet, puisque chaque valeur de $x^*(n)$ est maintenue constante durant la quantification, nous pouvons considérer que la fonction $x^*(n)$ n'est plus une fonction continue du temps mais une fonction de temps discret. Cela revient à nous placer dans le cas d'un échantillonnage idéal où $\tau \ll T_{ech}$ (cf. Figure 1 - 58).

Nous en déduisons donc que : s'il est placé devant un quantificateur, l'échantillonneur bloqueur n'a donc aucun effet sur le spectre du signal. Pour cette raison, et bien que la notation concerne un signal simplement échantillonné, nous noterons $x^*(n)$ le signal issu de l'échantillonneur bloqueur.

III.1.2. Le quantificateur

Quantifier une valeur de $x(t)$ c'est l'approximer grâce à un multiple entier d'une quantité élémentaire Δ dite « pas de quantification ». Le signal qui en résulte est noté $\hat{x}(t)$. Physiquement, bien qu'il soit généralement représenté comme un signal de valeur $n \times \Delta$ (ou $(n+1/2) \times \Delta$), $\hat{x}(t)$ est en fait un signal codé sur des bits. Ce code est appelé code thermomètre.

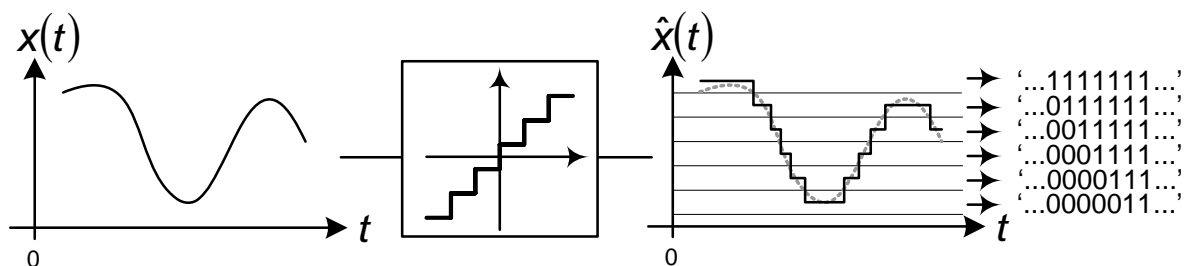


Figure 1 - 15 : Quantification et illustration du code thermomètre

La figure ci-dessous présente deux types de quantificateurs 8 niveaux (code thermomètre sur 7 bits).

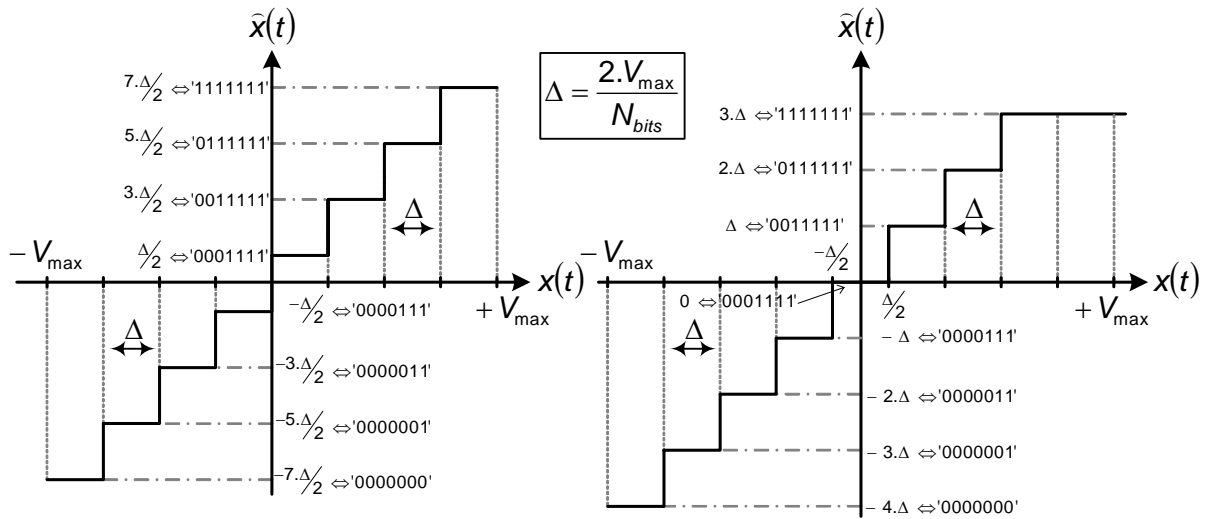


Figure 1 - 16 : Quantificateurs 8 niveaux de type « mid-riser » et « mid-tread »

La quantification induit une erreur dite « erreur de quantification », notée $e(t)$, et qui est définie par :

$$e(t) = \hat{x}(t) - x(t) \tag{Eq. 1 - 1}$$

La valeur de cette erreur de quantification (ici un quantificateur de type « mid-riser ») est comprise entre $\pm \Delta/2$ comme l'illustre la figure suivante [Cou96] :

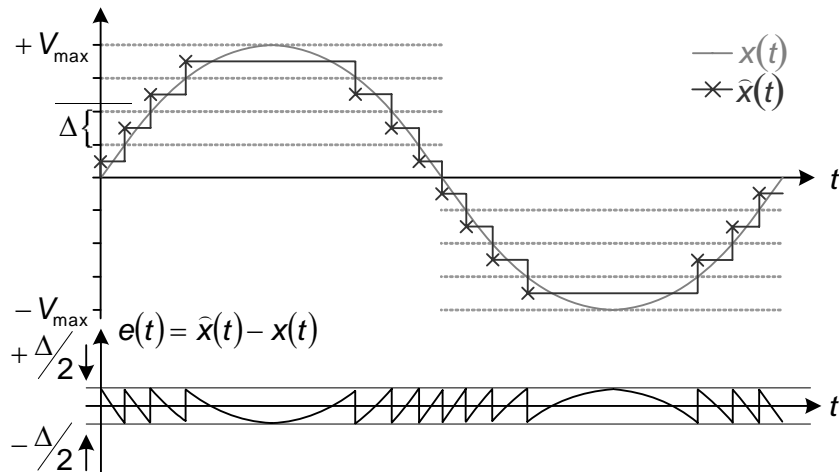


Figure 1 - 17 : Représentation de l'erreur de quantification

Nous pouvons dès lors assimiler l'erreur de quantification à une variable aléatoire de répartition uniforme dans l'intervalle $[- \Delta/2, + \Delta/2]$ [SrS77].

La puissance P_e de l'erreur s'écrit alors :

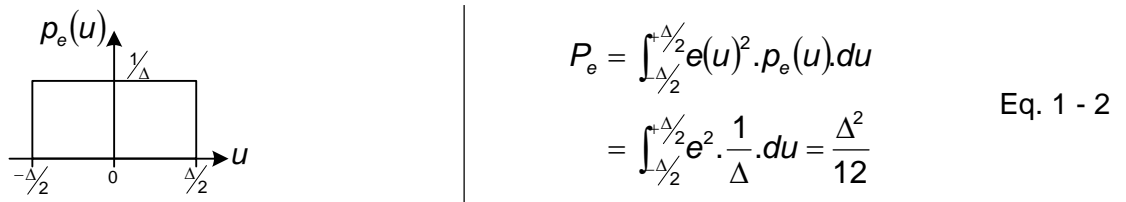


Figure 1 - 18 : Puissance du bruit de quantification

$p_e(u)$ désigne la densité de probabilité de la variable aléatoire $e(t)$.

Le code thermomètre n'étant pas optimum concernant l'encombrement, l'étape suivante de la conversion analogique/numérique consiste généralement à mettre en forme le code thermomètre $\hat{x}(t)$ en un code binaire $\hat{x}_c(t)$.

III.1.3. Le convertisseur analogique/numérique

Le convertisseur analogique/numérique peut être représenté par la figure ci-dessous. Il se compose d'un échantillonneur bloqueur, d'un quantificateur $2^{N_{bits}}$ niveaux et généralement d'un encodeur numérique N_{bits} :

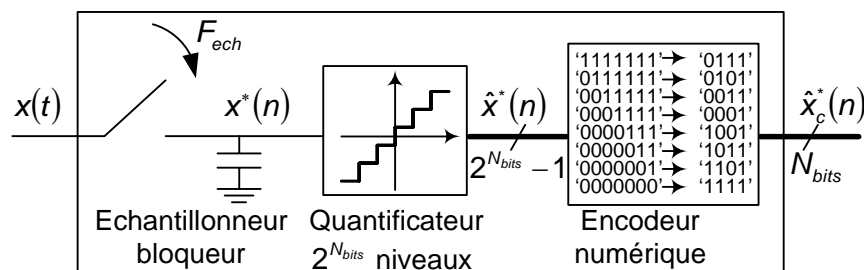


Figure 1 - 19 : Représentation fonctionnelle d'un convertisseur analogique/numérique

N.B. : les signaux $\hat{x}^*(n)$ et $\hat{x}_c^*(n)$ sont l'expression de la même information codée différemment, nous n'étudierons donc que le signal $\hat{x}^*(n)$.

Conformément à la théorie développée dans le chapitre suivant (cf. Chapitre II), les variables aléatoires de répartition uniforme engendrent un bruit de type blanc, c'est-à-dire de Densité Spectrale de Puissance (DSP) constante vis-à-vis des fréquences. Le bruit de quantification $e(t)$ est donc un bruit blanc, nous noterons sa DSP : $|E(f)|^2$. Pour calculer la valeur de $|E(f)|^2$, nous considérons que

la condition de Shannon est remplie ($F_{ech} \geq 2.f_b$), nous pouvons donc réduire le domaine d'étude à l'intervalle de fréquences $[-F_{ech}/2, F_{ech}/2]$:

$$P_e = \int_{-F_{ech}/2}^{+F_{ech}/2} |E(f)|^2 df$$

$$= |E(f)|^2 \cdot F_{ech}$$

Eq. 1 - 3

Soit :

$$|E(f)|^2 = \frac{\Delta^2}{12 \cdot F_{ech}}$$

Eq. 1 - 4

La répartition de la DSP sur l'intervalle de fréquences d'étude est décrite ci-dessous :

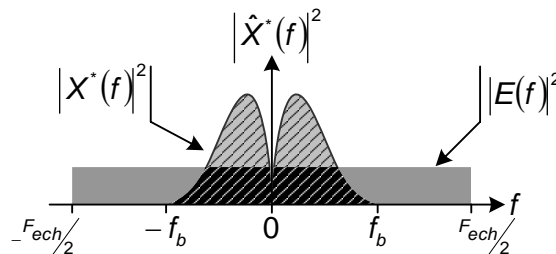


Figure 1 - 20 : DSP du bruit de quantification $e(t)$

$|X^*(f)|^2$ et $|\hat{X}^*(f)|^2$ désignent respectivement les spectres en puissance du signal échantillonné et du signal quantifié.

La puissance de bruit de quantification associé au signal $x(t)$ dans la bande utile $[-f_b, f_b]$ est alors :

$$P_{e \text{ sur } [-f_b, f_b]} = \int_{-f_b}^{f_b} |E(f)|^2 \cdot df$$

$$= \frac{2 \cdot f_b \cdot \Delta^2}{12 \cdot F_{ech}}$$

Eq. 1 - 5

De cette équation, nous déduisons que pour réduire le bruit dans la bande utile, il faudrait :

- soit augmenter la précision du quantificateur ce qui réduirait le pas de quantification : l'ajout d'un bit de précision divise la valeur de Δ par un facteur 2, soit une diminution de puissance du bruit en décibels de $-6.02dB$:

$$\Delta_{(N_{bits}+1)} = \frac{2 \cdot V_{\max}}{2^{N_{bits}+1}} = \frac{\Delta_{N_{bits}}}{2}$$

Eq. 1 - 6

- soit augmenter la fréquence d'échantillonnage : en multipliant par 2 la fréquence d'échantillonnage, le gain sur la puissance du bruit en décibels est de $-3.01dB$ soit l'équivalent d'un demi-bit.

Il est donc plus avantageux d'augmenter la résolution du quantificateur. Malheureusement la diminution du pas de quantification est limité par la technologie et l'architecture du quantificateur. On peut donc utiliser des systèmes utilisant une fréquence d'échantillonnage largement supérieure à $2.f_b$ (fréquence de Nyquist) : c'est la technique dite de « sur-échantillonnage ».

Le facteur de sur-échantillonnage, noté OSR (Over Sampling Ratio), peut alors être défini par [Son00]:

$$OSR = \frac{F_{ech}}{2.f_b} \quad \text{Eq. 1 - 7}$$

Le sur-échantillonnage est illustré ci-dessous :

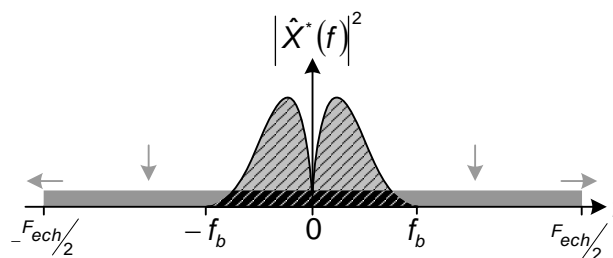


Figure 1 - 21 : Sur-échantillonnage du signal $x(t)$

Cependant, une augmentation de la fréquence d'échantillonnage pose des problèmes de consommation ou de limitation de la fréquence de commutation des interrupteurs MOS. Il faut donc se tourner vers des techniques de « mise en forme » du bruit de quantification (ou Noise Shaping) telles que les modulations Delta ou Sigma Delta qui font l'objet d'une étude dans les paragraphes suivants.

III.2. De la modulation Delta à la modulation Sigma Delta

La modulation Sigma Delta est historiquement une extension de la modulation Delta [Ste75]. Seule la modulation Sigma Delta est encore utilisée aujourd'hui pour « mettre en forme » le bruit de quantification lorsqu'elle est associée à des filtres décimateurs dans le cadre de la conversion analogique/numérique (cf. Figure 1 - 22). Les traitements numériques, post modulation, permettent la mise en forme des bits, le filtrage du signal utile ainsi que l'abaissement de la fréquence de travail, car les modulateurs sont de systèmes sur-échantillonnés. Par la suite, nous ne nous intéresserons qu'au fonctionnement des modulateurs, c'est-à-dire qu'au traitement analogique du signal.

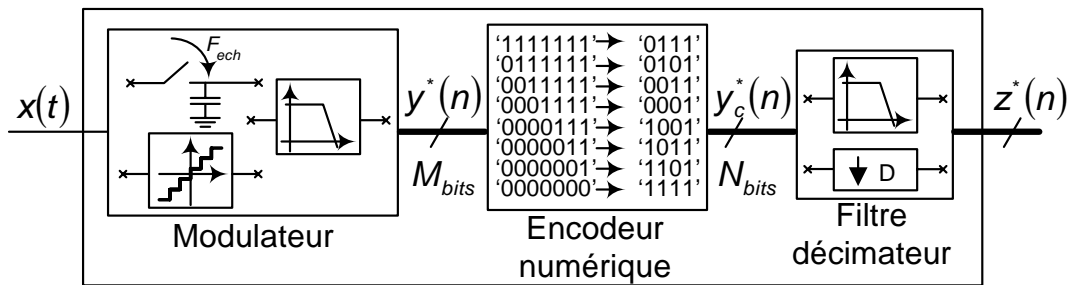


Figure 1 - 22 : Principe des convertisseurs analogique/numérique utilisant des modulateurs

Le principe commun aux deux modulations est le suivant : la quantification porte non pas sur la valeur absolue du signal $x(t)$ à un instant t_n , mais sur la différence entre ce signal et une valeur prédite $x_p(t)$ obtenues par extrapolation des valeurs précédentes. Le système travaille alors sur des quantités plus petites, le quantificateur peut donc avoir une dynamique $(2.V_{max})$ plus petite, ce qui permet de réduire le nombre de bits en sortie tout en conservant la précision globale du système.

La modulation Delta [Ste75] est apparue à un moment où les convertisseurs classiques étaient techniquement peu fiables ou économiquement non rentables. Les concepteurs ont alors mis au point cette modulation à partir de circuits simples. Le schéma bloc d'un modulateur Delta est représenté ci-dessous [Par90-2] :

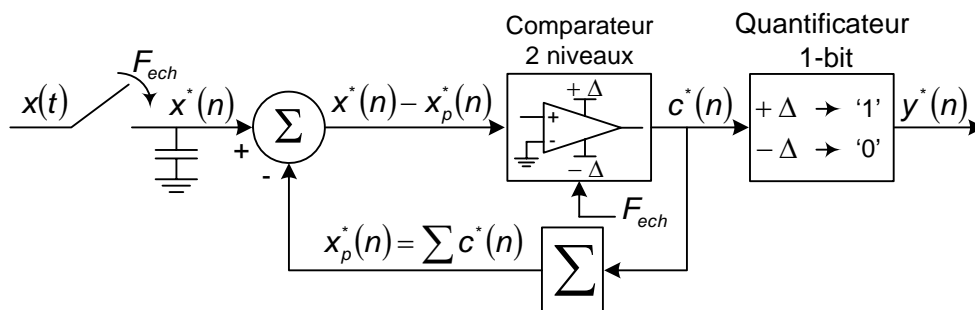


Figure 1 - 23 : Modulateur Delta

Le modulateur présenté est composé d'un échantillonneur bloqueur, d'un soustracteur, d'un comparateur 2 niveaux $(\pm \Delta)$, d'un sommateur qui joue le rôle de prédicteur des valeurs de $x^*(n)$ et d'un quantificateur 1 bit qui permet de convertir les signaux issus du comparateur en signaux codés sur un bit. Ce type de modulateur ne nécessite pas d'encodeur numérique.

Le fonctionnement du modulateur est décrit par la figure ci-dessous [Auv00] : à chaque période d'échantillonnage T_{ech} , le signal d'entrée $x^*(n)$ est comparé avec sa valeur prédite $x_p^*(n)$. Si la différence entre ces deux valeurs est positive alors la sortie du comparateur $c^*(n)$ vaut $+\Delta$, le

signal $x_p^*(n)$ est alors augmenté d'une quantité Δ et la sortie du modulateur $y^*(n)$ vaut '1'. Dans le cas contraire, le signal $x_p^*(n)$ est diminué de la même quantité Δ et $y^*(n)$ vaut '0'.

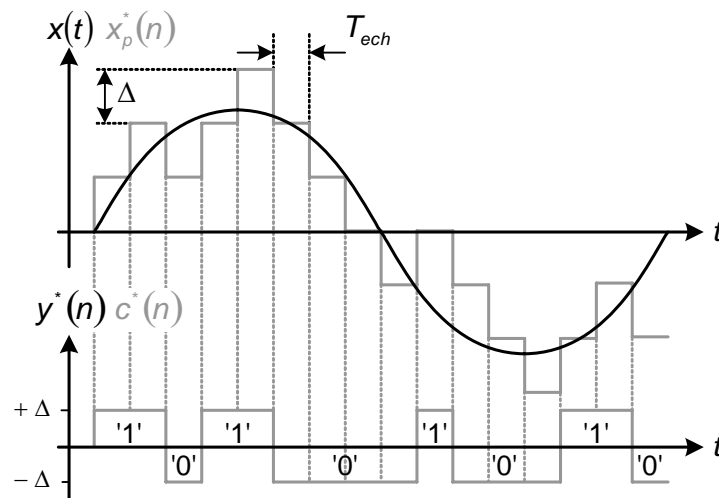


Figure 1 - 24 : Modulation Delta

Un facteur limitant de la modulation Delta provient de la saturation de pente. En effet, pour que la quantification soit réaliste, il faut éviter que la variation du signal n'excède la pente autorisée par le système, soit pour un signal entrant $x_o(t) = A \cdot \cos(2 \cdot \pi \cdot f_o \cdot t)$:

$$A \cdot 2 \cdot \pi \cdot f_o \leq \frac{\Delta}{T_{ech}} \quad \text{Eq. 1 - 8}$$

Ce qui équivaut à limiter l'amplitude du signal d'entrée lorsque sa fréquence est fixée, ou inversement, limiter la fréquence du signal d'entrée pour une amplitude fixe.

La modulation Sigma Delta est née de cette constatation. Afin de réduire le phénomène de saturation de pente, le signal est intégré avant sa comparaison avec le signal prédit (comme le signal est discrétisé, nous utiliserons un sommateur). La condition de saturation de pente devient alors :

$$A \leq \frac{\Delta}{T_{ech}} \quad \text{Eq. 1 - 9}$$

Elle ne dépend donc plus de la fréquence du signal d'entrée, il suffit alors de prendre le pas du comparateur au moins égal à l'amplitude maximum pour éviter toute risque de saturation.

N.B. : Une autre explication plausible concernant la « naissance » de la modulation Sigma Delta, vient du fait que la modulation Delta code la « dérivée » du signal $(x^*(n) - x_p^*(n)) / T_{ech}$ et non le signal, d'où l'intégration en amont qui permet de récupérer en sortie le signal modulé.

Le schéma bloc devient alors :

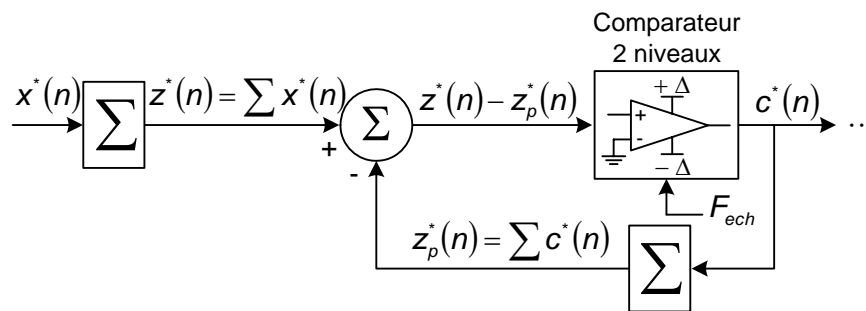


Figure 1 - 25 : Modulateur Delta précédé d'un sommateur

Comme l'intégration est une opération linéaire, il est possible de faire « entrer » les intégrateurs (les sommateurs) dans la boucle après le soustracteur. Nous obtenons alors un modulateur Sigma Delta :

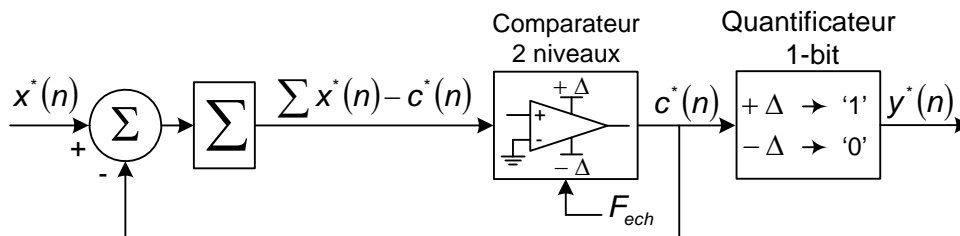


Figure 1 - 26 : Modulateur Sigma Delta (1 bit)

N.B. : Le nom de la modulation Sigma Delta vient de l'intégration (Σ) du signal avant sa modulation Delta (Δ).

III.3. Les modulateurs Sigma Delta

Malgré son introduction dès 1962 par Inose, Yasuda et Markami [IYM62], les modulateurs Sigma Delta ont longtemps été confinés dans des architectures d'ordre au plus égal à deux (une ou deux boucles de contre-réaction) à cause des risques d'instabilité des architectures d'ordre supérieur. Ils n'ont réellement pris de l'importance qu'à partir de la fin des années 80 lorsque Lee et Sodini ont mis au point une technique permettant d'accroître l'ordre des modulateurs [LeS87] [Ger96].

Comme nous l'avons dit précédemment (cf. §III.1), la quantification n'est pas une opération instantanée, il faut donc placer un échantillonneur bloqueur en amont du quantificateur. Suivant la position de l'échantillonneur bloqueur, avant ou après l'intégrateur (ou sommateur) nous parlerons de modulateur Sigma Delta à temps discret ou de modulateur Sigma Delta à temps continu. Nous ne nous intéresserons qu'aux modulateurs Sigma Delta à temps discret, comme celui présenté Figure 1 - 26.

De plus, pour accroître la précision des modulateurs Sigma Delta le comparateur 2 niveaux de la Figure 1 - 26 est généralement remplacé par un quantificateur $2^{N_{bits}}$ niveaux étudié précédemment (cf. §III.1.2). L'introduction du quantificateur dans le modulateur entraîne quelques aménagements, puisque la sortie de ce dernier est codée sur $2^{N_{bits}} - 1$ bits il faut remettre sous forme analogique ce code thermomètre afin qu'il soit exploitable dans la contre-réaction. Cette mise en forme est réalisée par un convertisseur numérique/analogique interne (à ne pas confondre avec une chaîne complète de conversion) et donne le signal $x_p^*(n)$, qui est toujours valeur prédite du signal d'entrée $x^*(n)$. La figure ci-dessous donne un exemple de réalisation :

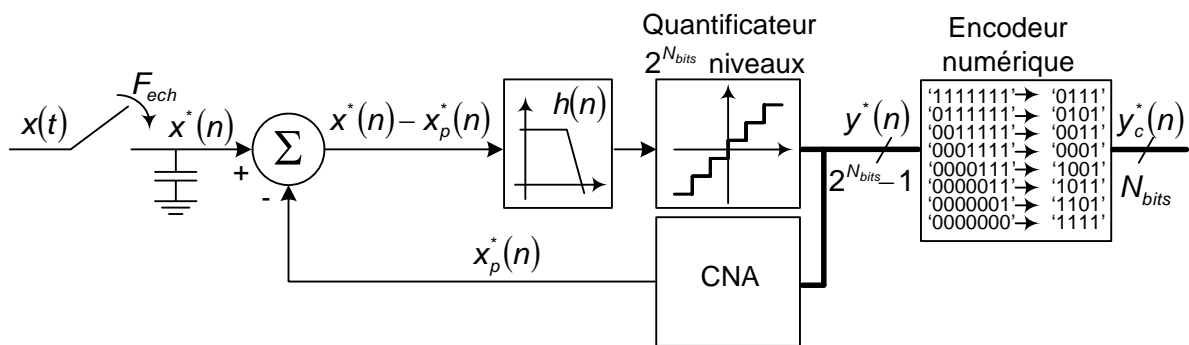


Figure 1 - 27 : Modulateur Sigma Delta du premier ordre avec un quantificateur $2^{N_{bits}}$ niveaux

III.3.1. Les modulateurs Sigma Delta passe bas : théorie générale

Comme cela vient d'être énoncé dans l'introduction, le modulateur Sigma Delta a longtemps été employé avec des architectures du premier ordre avec un filtre $h(n)$ de type passe bas. Nous partons donc de ce premier modèle pour définir la théorie de base qui régit la « mise en forme » du bruit de quantification (Noise Shaping). Nous étudierons ensuite deux autres architectures de modulateurs Sigma Delta passe bas : une du second ordre et une autre à structure MASH [MUI97].

III.3.1.1. Modulateur du premier ordre.

Les performances du modulateur Sigma Delta se mesurent dans le domaine fréquentiel, et comme nous avons affaire à un système échantillonné nous prendrons la transformée en z de l'ensemble des signaux. Ainsi nous noterons $X^*(z)$, $X_p^*(z)$, $Y^*(z)$ et $H(z)$, les transformées en z respectives des signaux $x^*(n)$, $x_p^*(n)$, $y^*(n)$ et du filtre $h(n)$.

Afin de faciliter les calculs, le CNA est généralement remplacé par un retard pur z^{-1} et le quantificateur par une source additive de bruit blanc $e^*(n)$ ([Ben48], §III.1.2) de transformée en z

notée $E^*(z)$. Ainsi le modulateur Sigma Delta précédemment décrit (cf. Figure 1 - 27) peut se modéliser de la façon suivante :

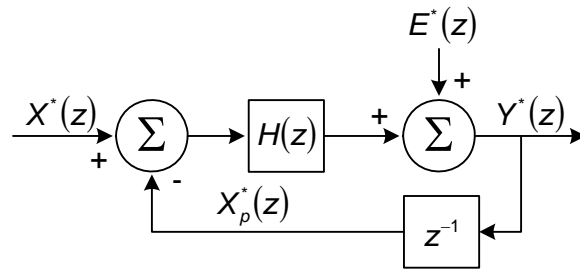


Figure 1 - 28 : Transformée en z des signaux d'un modulateur Sigma Delta du premier ordre

L'expression de la fonction de transfert liée à ce type de montage est la suivante :

$$Y^*(z) = X^*(z) \cdot \frac{H(z)}{1 + z^{-1} \cdot H(z)} + E^*(z) \cdot \frac{1}{1 + z^{-1} \cdot H(z)} \quad \text{Eq. 1 - 10}$$

Cette fonction peut alors se décomposer en deux fonctions de transfert : la STF (Signal Transfer Function) et la NTF (Noise Transfer Function) :

$$STF(z) = \left. \frac{Y^*(z)}{X^*(z)} \right|_{E^*(z)=0} = \frac{H(z)}{1 + z^{-1} \cdot H(z)} \quad \text{Eq. 1 - 11}$$

$$NTF(z) = \left. \frac{Y^*(z)}{E^*(z)} \right|_{X^*(z)=0} = \frac{1}{1 + z^{-1} \cdot H(z)}$$

Considérons maintenant le cas particulier où le filtre est un intégrateur de formule :

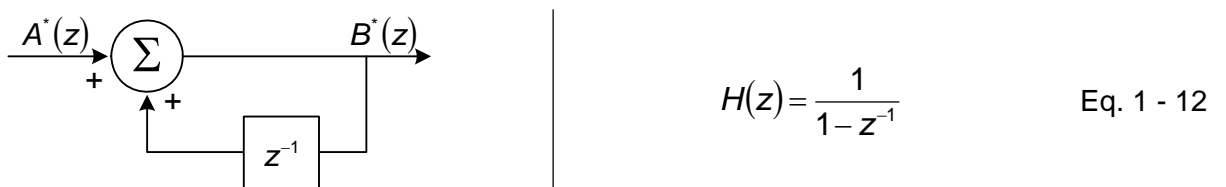


Figure 1 - 29 : Filtres $H(z)$

Alors les fonctions de transfert STF et NTF :

$$STF(z) = 1$$

$$NTF(z) = 1 - z^{-1} \quad \text{Eq. 1 - 13}$$

En remplaçant z^{-1} par sa valeur $z^{-1} = e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_{ech}}}$ nous trouvons :

$$|STF(f)|^2 = 1$$

$$|NTF(f)|^2 = \left(1 - e^{j \cdot \frac{2 \cdot \pi \cdot f}{F_{ech}}} \right) \left(1 - e^{-j \cdot \frac{2 \cdot \pi \cdot f}{F_{ech}}} \right) = 4 \cdot \sin^2 \left(\frac{\pi \cdot f}{F_{ech}} \right) \quad \text{Eq. 1 - 14}$$

Le signal n'est pas affecté et le bruit est « mis en forme » : les basses fréquences sont atténuées, on dit alors que le bruit est rejeté en haute fréquence. La figure suivante illustre ce phénomène :

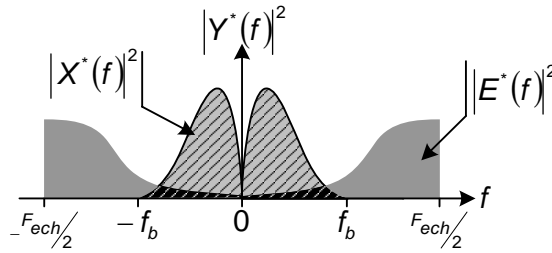


Figure 1 - 30 : Mise en forme du bruit (Noise Shaping)

Ainsi la puissance du bruit dans la bande utile $[-f_b, f_b]$ est alors :

$$\begin{aligned}
 P_{e \text{ sur } [-f_b, f_b]} &= \int_{-f_b}^{f_b} |E(f)|^2 \cdot |NTF(f)|^2 df \\
 &= \frac{\Delta^2}{12 \cdot F_{ech}} \cdot 4 \cdot \left(f_b - \frac{F_{ech}}{2 \cdot \pi} \cdot \sin\left(\frac{2 \cdot \pi \cdot f_b}{F_{ech}}\right) \right) \\
 &\approx \frac{\Delta^2}{12} \cdot \frac{\pi^2}{3} \cdot \left(\frac{2 \cdot f_b}{F_{ech}}\right)^3 \quad \text{avec } 2 \cdot f_b \ll F_{ech}
 \end{aligned}
 \tag{Eq. 1 - 15}$$

Pour augmenter la bande utile ou tout simplement diminuer le plancher de bruit à bande utile constante, nous aurons alors recours à des modulateurs d'ordre supérieur.

III.3.1.2. Modulateur du second ordre et généralisation

Considérons l'exemple d'un modulateur d'ordre 2 :

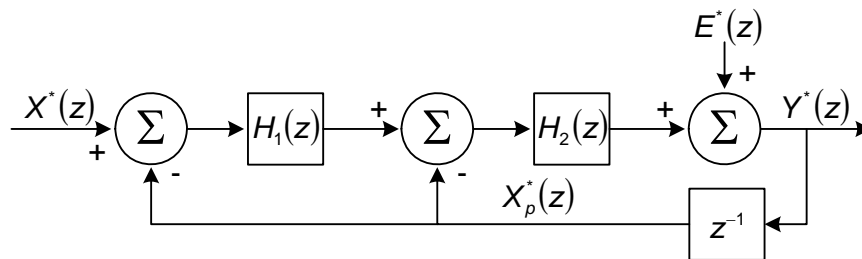


Figure 1 - 31 : Modulateur Sigma Delta du second ordre

Les fonctions de transfert de ce type de montage sont alors :

$$\begin{aligned}
 STF(z) &= \left. \frac{Y^*(z)}{X^*(z)} \right|_{E^*(z)=0} = \frac{H_1(z) \cdot H_2(z)}{1 + z^{-1} \cdot H_2(z) + z^{-1} \cdot H_1(z) \cdot H_2(z)} \\
 NTF(z) &= \left. \frac{Y^*(z)}{E^*(z)} \right|_{X^*(z)=0} = \frac{1}{1 + z^{-1} \cdot H_2(z) + z^{-1} \cdot H_1(z) \cdot H_2(z)}
 \end{aligned}
 \tag{Eq. 1 - 16}$$

Considérons maintenant le cas particulier où les filtres sont des intégrateurs de formules :

$$H_1(z) = H_2(z) = \frac{1}{1 - z^{-1}} \quad \text{Eq. 1 - 17}$$

Les fonctions de transfert STF et NTF sont alors :

$$\begin{aligned} STF(z) &= 1 \\ NTF(z) &= (1 - z^{-1})^2 \end{aligned} \quad \text{Eq. 1 - 18}$$

En remplaçant z^{-1} par sa valeur $z^{-1} = e^{-j \frac{2\pi f}{F_{ech}}}$ nous trouvons ;

$$\begin{aligned} |STF(f)|^2 &= 1 \\ |NTF(f)|^2 &= \left(4 \cdot \sin^2 \left(\frac{\pi \cdot f}{F_{ech}} \right) \right)^2 \end{aligned} \quad \text{Eq. 1 - 19}$$

La puissance du bruit dans la bande utile $[-f_b, f_b]$ est alors :

$$P_{e \text{ sur } [-f_b, f_b]} \approx \frac{\Delta^2}{12} \cdot \frac{\pi^4}{5} \cdot \left(\frac{2 \cdot f_b}{F_{ech}} \right)^5 \quad \text{avec} \quad 2 \cdot f_b \ll F_{ech} \quad \text{Eq. 1 - 20}$$

De manière générale, les fonctions de transfert d'un modulateur Sigma Delta d'ordre n s'écrivent :

$$\begin{aligned} |STF(f)|^2 &= 1 \\ |NTF(f)|^2 &= 2^{2 \cdot n} \cdot \sin^{2 \cdot n} \left(\frac{\pi \cdot f}{F_{ech}} \right) \end{aligned} \quad \text{Eq. 1 - 21}$$

En approximant le sinus par le premier terme de son développement limité, la puissance du bruit dans la bande utile $[-f_b, f_b]$ est alors :

$$P_{e \text{ sur } [-f_b, f_b]} \approx \frac{\Delta^2}{12} \cdot \frac{\pi^{2 \cdot n}}{2 \cdot n + 1} \cdot \left(\frac{2 \cdot f_b}{F_{ech}} \right)^{2 \cdot n + 1} \quad \text{Eq. 1 - 22}$$

Ainsi, en augmentant l'ordre des $\Sigma\Delta$ le bruit est théoriquement rejeté de plus en plus loin de la bande utile, comme l'illustre les courbes suivantes :

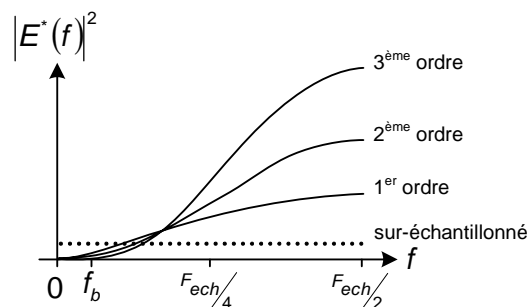


Figure 1 - 32 : Mise en forme du bruit pour les modulateurs d'ordre 1, 2, et 3

Théoriquement il n'y a pas de limitation à l'augmentation de l'ordre des modulateurs Sigma Delta ; néanmoins des problèmes de stabilité, viennent compliquer l'utilisation de modulateurs d'ordres supérieurs à 2 [NSA97].

III.3.1.3. Structure MASH

L'apparition des structures MASH (MultistAge noise SHaper) date de 1987 [MUI97] [Rib91]. Ce type d'architecture a été créé par la mise en cascade de structures définies précédemment, ce qui permet d'accroître la mise en forme du bruit sans toutefois rencontrer les problèmes de stabilité des structures linéaires. Un exemple d'un tel modulateur est décrit ci-dessous, il comporte deux boucles du premier ordre mises en cascade :

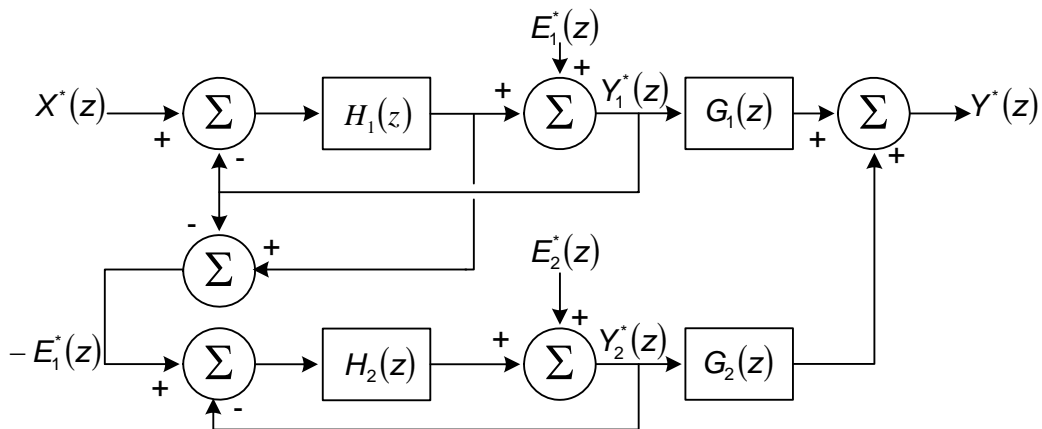


Figure 1 - 33 : Structure MASH

La première boucle fonctionne de manière classique : $X^*(z)$ le signal d'entrée est filtré puis quantifié. Le bruit de quantification $E_1(z)$ de la première boucle est ensuite inversé et mis en forme à son tour, et c'est là la particularité de ce montage. Les sorties des deux boucles sont ensuite filtrées (en général par un retard) puis sommées.

La fonction de transfert d'un tel montage est la suivante :

$$Y^*(z) = X^*(z) \cdot \frac{G_1(z)H_1(z)}{1+H_1(z)} + E_1(z) \left(\frac{G_1(z)}{1+H_1(z)} - \frac{H_2(z)G_2(z)}{1+H_2(z)} \right) + E_2(z) \cdot \frac{G_2(z)}{1+H_2(z)} \quad \text{Eq. 1 - 23}$$

Grâce à un choix pertinent des filtres, nous pouvons alors faire totalement disparaître l'effet du bruit de quantification $E_1(z)$ au prix d'une mise en forme plus importante du bruit de quantification de la seconde boucle, $E_2(z)$.

Considérons l'exemple suivant [ASS96] :

$$H_1(z) = H_2(z) = \frac{z^{-1}}{1 - z^{-1}}, G_1 = z^{-1} \text{ et } G_2 = (1 - z^{-1}) \quad \text{Eq. 1 - 24}$$

Les fonctions de transfert STF et NTF sont alors :

$$\begin{aligned} STF(z) &= z^{-2} \\ NTF(z) &= (1 - z^{-1})^2 \end{aligned} \quad \text{Eq. 1 - 25}$$

Soit une mise en forme du bruit identique à celle obtenue avec un modulateur du second ordre, alors que seules deux boucles du premier ordre sont mises en œuvre. C'est l'intérêt des structures MASH, qui permettent d'obtenir des structures stables d'ordre élevé à partir de boucles du premier ou second ordre. La grande différence provient du double retard (z^{-2}) appliqué sur le signal d'entrée.

N.B. : Il est possible de former des structures cascadées plus complexes, notamment en augmentant le nombre de boucles et leur ordre [Ded94].

III.3.2. Les modulateurs Sigma Delta passe bande : notre application

Les modulateurs Sigma Delta que nous venons d'étudier sont utilisés pour la conversion en bande de base, c'est à dire lorsque la fréquence de travail (ou de conversion) est proche ou égale à 0Hz. Ce type de conversion ne peut s'appliquer à la chaîne réception RF décrite précédemment (cf. §II.3). En effet, en considérant le rappel effectué par la figure ci-dessous, nous constatons que la fréquence intermédiaire f_{i_2} est trop éloignée de la fréquence nulle pour pouvoir utiliser un modulateur Sigma Delta passe bas. Nous devons donc nous tourner vers des modulateurs travaillant à des fréquences plus élevées : ce sont les modulateurs passe bande.

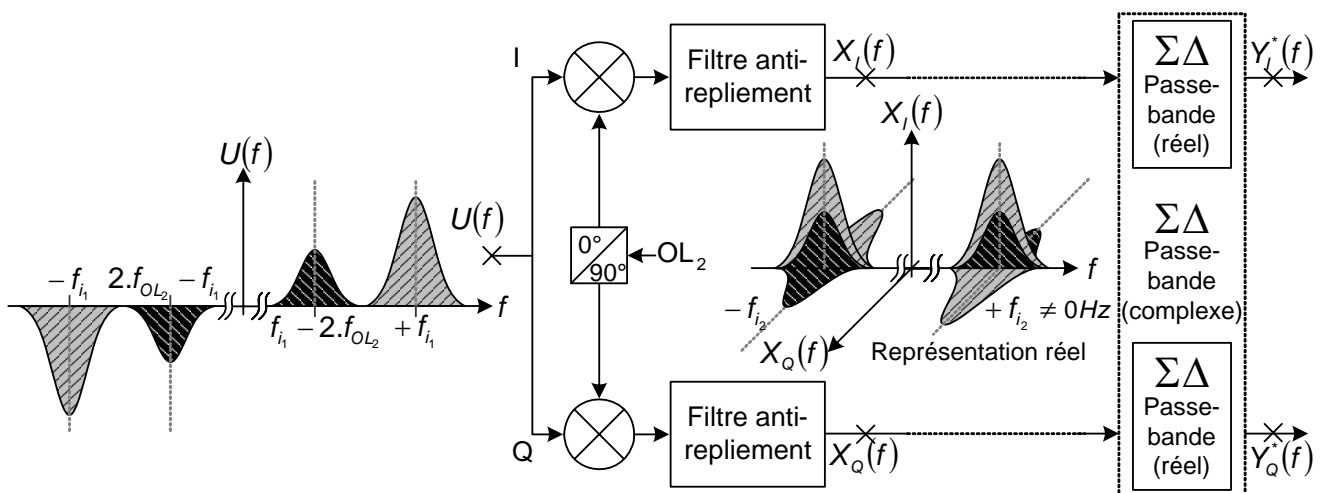


Figure 1 - 34 : Modulateur Sigma Delta passe bande dans la chaîne de réception RF

Le signal $U(f)$ représenté ci-dessus est issu de l'amplificateur CAG (cf. §II.3). Nous considérons que ce signal entrant dans le second mélangeur polyphasé est composé d'un signal utile à la fréquence intermédiaire f_{i_1} et d'un signal parasite à la fréquence $f_{i_1} - 2.f_{OL_2}$ (cette hypothèse est réaliste vu l'absence de filtre de canal avant le second mélangeur polyphasé). Après le filtre anti-repliement, nous obtenons alors deux signaux en phase sur la voie I et en opposition de phase sur la voie Q. Nous nous servirons de ces signaux entrant dans les modulateurs comme base pour comparer les deux types de modulateurs.

Nous parlerons de modulateurs réels, lorsque le traitement des voies en quadrature de phase I et Q s'effectue séparément par des modulateurs monovoie ; par opposition, nous désignerons par modulateurs complexes le traitement simultanément des deux voies.

N.B. : Le type de conversion employée n'a aucune influence sur le type de représentations utilisé pour décrire la sortie du modulateur. Nous utiliserons, pour notre part, une représentation complexe.

III.3.2.1. Les modulateurs Sigma Delta passe bande réels

Reprenons l'étude effectuée sur le modulateur Sigma Delta du premier ordre (cf. §III.3.1.1) et étendons la à un modulateur monovoie réel. Nous cherchons maintenant à déterminer les propriétés du filtre $H(z)$. Les performances recherchées du modulateur passe bande autour de la fréquence intermédiaire f_{i_2} sont les mêmes que pour un modulateur passe bas autour de la fréquence nulle ; c'est-à-dire la conservation du signal dans la bande utile et rejet du bruit hors de cette bande de fréquences :

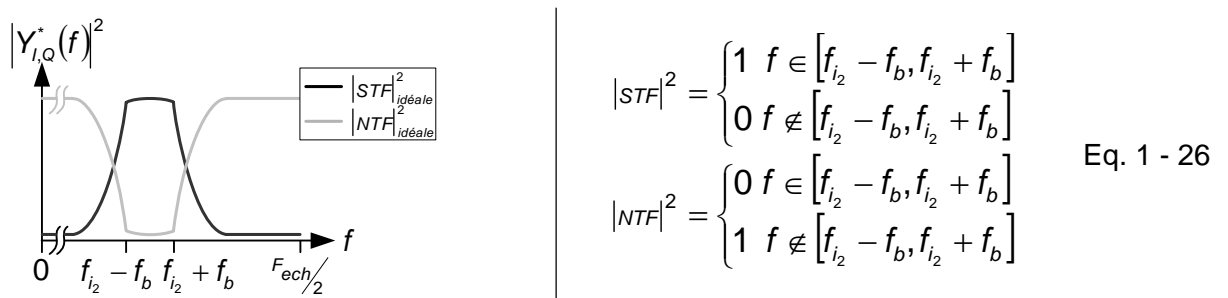


Figure 1 - 35 : Propriétés recherchées du modulateur Sigma Delta passe bande

N.B. : ces caractéristiques sont évidemment théoriques et très difficiles à obtenir. En réalité, la STF a des pentes beaucoup moins marquées et est non nulle en-dehors de la bande passante.

Dans ces conditions et en nous basant sur les fonctions de transfert d'un modulateur du premier ordre (cf. Eq. 1 - 11), le filtre doit alors avoir les caractéristiques suivantes :

$$|H(z)| = \begin{cases} \infty & f \in [f_{i_2} - f_b, f_{i_2} + f_b] \\ 0 & f \notin [f_{i_2} - f_b, f_{i_2} + f_b] \end{cases} \quad \text{Eq. 1 - 27}$$

Ce sont généralement des filtres passe bande ou des résonateurs centrés sur la fréquence f_{i_2} qui sont utilisés. Comme par définition le spectre d'un filtre réel (qui ne traite qu'une seule composante du signal) est symétrique par rapport à l'axe des fréquences nulles, les propriétés valables sur l'intervalle de fréquences $[f_{i_2} - f_b, f_{i_2} + f_b]$, le sont également sur $[-f_{i_2} - f_b, -f_{i_2} + f_b]$. Ainsi le spectre des modulateurs réels sera symétrique par rapport à l'axe des fréquences nulles comme cela est illustré par la Figure 1 - 36.

La conception de filtres passe bande ou de résonateurs ayant une large bande passante est parfois complexe et coûteuse. Dans ce cas, l'utilisation de modulateurs Sigma Delta d'ordre supérieur ou cascades permet d'élargir la bande utile du modulateur en introduisant alors de nouveaux pôles dans la fonction de transfert [JSS91]. La figure suivante illustre l'utilisation dans un modulateur du second ordre (cf. Figure 1 - 31) de deux résonateurs identiques dont la fréquence centrale f_{i_2} est décalée de $2.\varepsilon$:

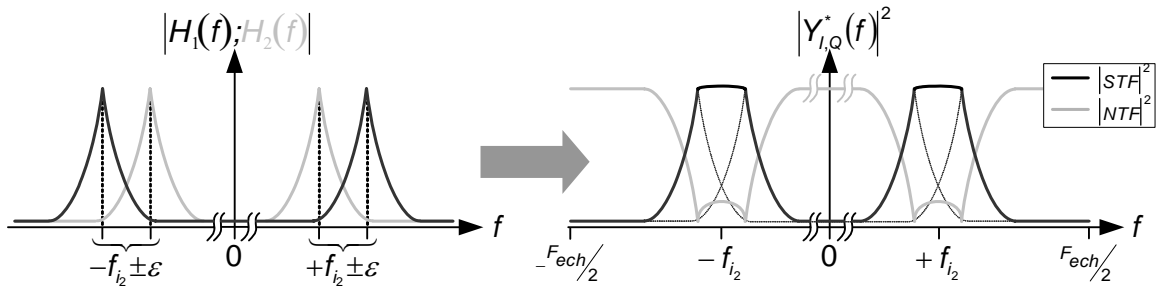


Figure 1 - 36 : Modulateur Sigma Delta passe bande du second ordre avec résonateurs

Nous utilisons maintenant ce type de modulateurs Sigma Delta monovoie pour moduler les signaux sortant du filtre anti-repliement (cf. Figure 1 - 34). Nous constatons alors que la mise en forme du bruit de quantification a été effective aux fréquences $+f_{i_2}$ et $-f_{i_2}$. Le signal parasite, tout comme le signal utile, est donc converti et susceptible de se replier dans la bande de base lors de traitements numériques ultérieurs.

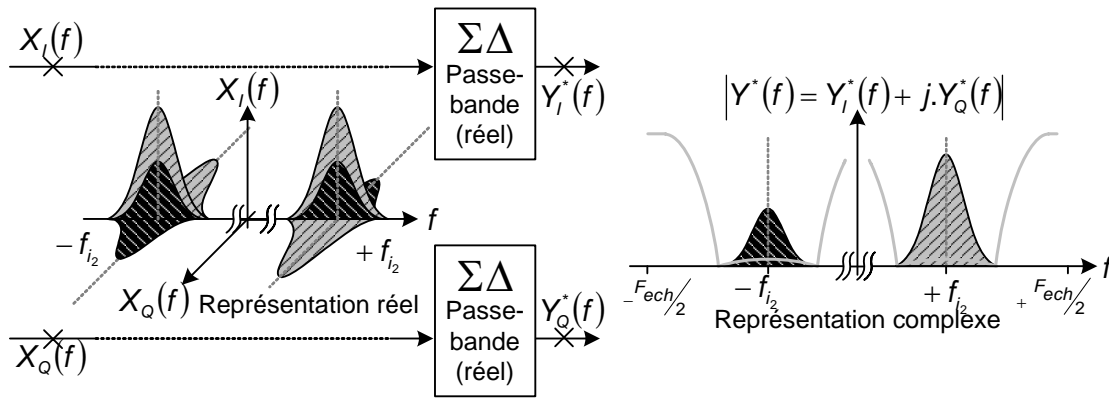


Figure 1 - 37 : Modulateurs Sigma Delta passe bande réels

L'idéal serait de ne convertir que le signal utile. C'est ce que réalise le modulateur Sigma Delta Complexe dont l'étude suit.

III.3.2.2. Les modulateurs Sigma Delta passe bande complexes

Les modulateurs Sigma Delta passe bande complexes utilisent des filtres complexes qui traitent simultanément les voies I et Q. La théorie concernant les filtres complexes est développée en Annexe 2 (cf. §I.2). La figure ci-dessous donne un exemple de modulateur du second ordre :

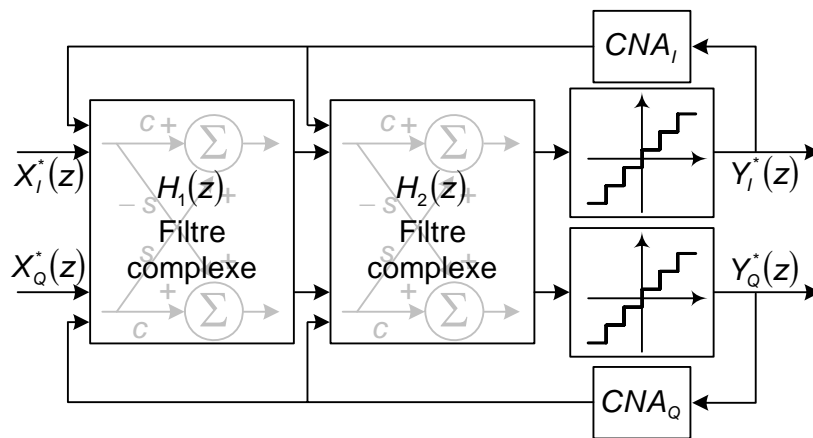


Figure 1 - 38 : Modulateur Sigma Delta passe bande complexe

Comme pour la structure réelle, l'utilisation de deux filtres (ici des résonateurs) centrés respectivement autour des fréquences $+f_{i_2} \pm \varepsilon$ permet d'élargir la bande passante du modulateur. Un modulateur Sigma Delta utilisant ce type de filtres complexes aura alors les fonctions de transfert qui le caractérisent (STF et NTF) semblables à celle illustrées ci-dessous :

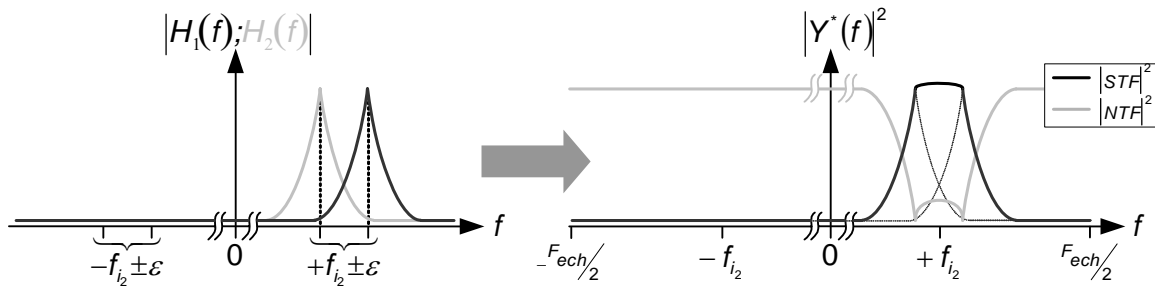


Figure 1 - 39 : STF et NTF d'un modulateur Sigma Delta passe bande complexe

Nous utilisons maintenant ce type de modulateur Sigma Delta complexe pour moduler les signaux sortant du filtre anti-repliement (cf. Figure 1 - 34). Le signal parasite est alors noyé dans le bruit de quantification :

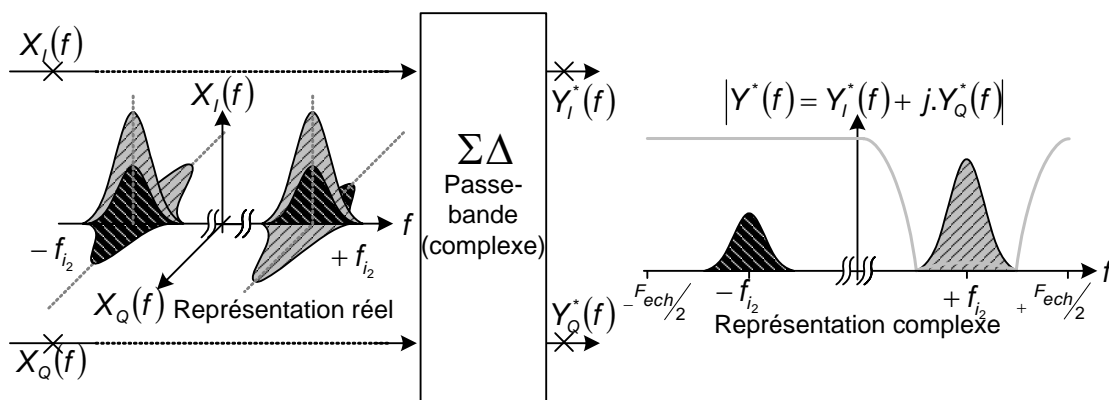


Figure 1 - 40 : Modulateur Sigma Delta passe bande complexe

La théorie des modulateurs Sigma Delta que nous venons de développer est indépendante de toute technologie. Cependant, comme énoncé dans l'introduction du paragraphe §III.3, ce type de modulateurs ne s'est réellement développé que lorsque les technologies CMOS ont été propices à une forte intégration des filtres. Les filtres que nous utiliserons par la suite sont des architectures à capacités commutées. Ce type d'architecture est particulièrement adapté aux modulateurs Sigma Delta temps discret que nous utilisons. Nous allons maintenant nous intéresser aux filtres utilisés dans les modulateurs.

IV. Les filtres à capacités commutées

Bien que l'utilisation généralisée de ce type de filtres soit récente - le premier filtre intégré utilisant des capacités commutées fut commercialisé en 1977 - le principe des capacités commutées était connu depuis longtemps. En effet on s'accorde pour octroyer la paternité du principe à Maxwell (1873) avec le « principe du courant intermittent » [Max73].

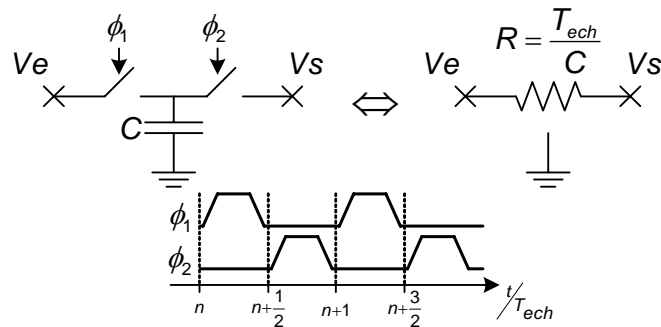


Figure 1 - 41 : Structure à capacités commutées équivalente à une résistance

L'intérêt des capacités commutées réside dans la simulation d'une résistance par une capacité et un jeu d'interrupteurs, évitant ainsi l'intégration de résistances diffusées [BBL96-1] : durant la phase ϕ_1 la charge de la capacité est égale à : $Q_{\phi_1} = C.V_e$. Durant la phase ϕ_2 la charge est égale à $Q_{\phi_2} = C.V_s$. A l'issu d'une période d'échantillonnage T_{ech} , le courant qui a transité de V_e vers V_s est alors égal à : $I_{es} = C.(V_e - V_s)/T_{ech}$. Ce qui est équivalent à une résistance de valeur : $R = T_{ech}/C$.

Cette approximation est réaliste si les entrées/sorties sont des générateurs de tension (faible impédance), si les régimes transitoires (charges/décharges des capacités) sont brefs et si les interrupteurs ne sont pas fermés simultanément (horloges non-recouvrantes).

C'est l'avènement et la pleine maîtrise des technologies MOS qui ont permis de répandre l'usage des capacités commutées, notamment dans le filtrage analogique. D'autant plus que ce type d'architecture est pleinement compatible avec les contraintes VLSI (Very Large Scale Integration), ce qui en fait un candidat sérieux pour les circuits ayant de fortes densités d'intégration, ainsi que pour les circuits mixtes.

IV.1. Rappel théorique et exemples

IV.1.1. Bases des filtres à capacités commutées

Le gain du montage présenté précédemment (Figure 1 - 41) dépend de la valeur de la capacité. Or il est technologiquement difficile d'obtenir des précisions inférieures à quelques pourcents ce qui est insuffisant. Comme nous allons l'expliquer l'utilisation d'un montage intégrateur permet de contourner cette difficulté.

Considérons le montage suivant, composé d'une structure simple à capacité commutée, d'un Amplificateur Opérationnel (souvent à Transconductance (A.O.T.)) et d'une capacité de retour (rétroaction). Par convention, nous considérons, dans un premier temps, que le signal d'entrée V_e est échantillonné aux instants $n.T_{ech}$ (phase ϕ_2) et bloqué durant T_{ech} et le signal de sortie V_s est également « lu » aux mêmes instants $n.T_{ech}$:

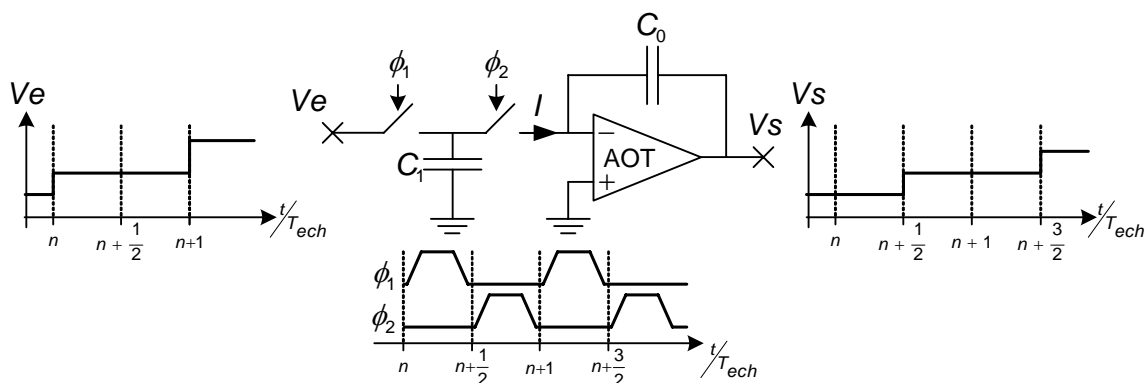


Figure 1 - 42 : Principe de l'intégrateur à capacités commutées

En utilisant l'équivalence avec la résistance :

$$\left. \begin{array}{l} V_{e-0} \approx I \cdot \frac{T_{ech}}{C_1} \\ V_{s-0} \approx -\frac{I}{C_0 \cdot p} \end{array} \right\} \Rightarrow V_s \approx -V_e \cdot \frac{C_1}{T_{ech} \cdot C_0} \cdot \frac{1}{p} \quad \text{Eq. 1 - 28}$$

La fonction de transfert dépend alors du rapport de deux capacités de même nature, ce qui accroît la précision du gain du montage. Le gain est alors indépendant du procédé technologique.

Une autre contrainte, cette fois-ci économique, vient s'ajouter à la réalisation des architectures à capacités commutées. La surface de silicium étant « coûteuse », il est intéressant d'avoir des capacités de faible surface et donc de faible valeur. Or plus la valeur est faible et moins les capacités

parasites (cf. §IV.2) peuvent être négligées. Il faut donc changer d'architecture pour se tourner vers des structures moins sensibles aux parasites (cf. Annexe 3 §I.3).

Nous allons maintenant dresser une liste non exhaustive des différentes « briques de base » qui sont le plus couramment utilisées, ainsi que les fonctions de transfert qui leur sont associées [Bas00].

IV.1.2. Structures avec rétroaction capacitive

Nous considérons maintenant que le signal d'entrée n'est plus échantillonné au préalable, mais que cette opération est réalisée par la structure. Le montage ci-dessous réalise les opérations de sommation, d'intégration négative, positive et de multiplication par une constante :

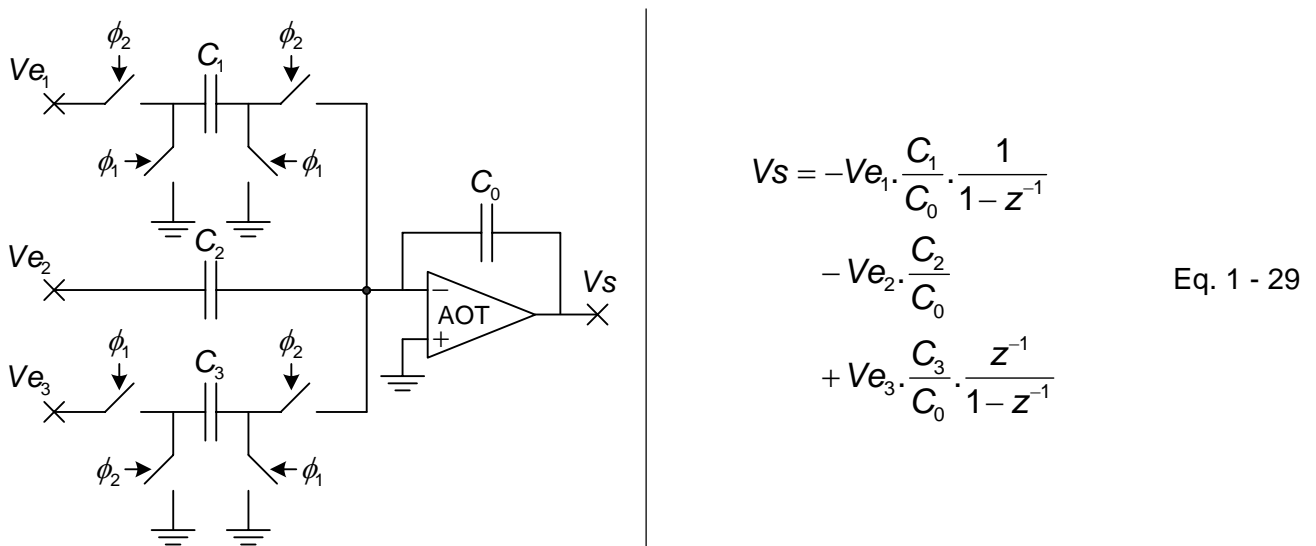


Figure 1 - 43 : Cellules de base avec simple capacité de feed-back

Le calcul de la fonction de transfert s'effectue en utilisant les transferts de charges entre les capacités, soit :

$$\Delta Q_{C_0} \left(n + \frac{1}{2} \right) = C_0 \cdot \left(\underbrace{V_s \left(n + \frac{1}{2} \right)}_{=V_s(n+1)} - V_s(n) \right) = \Delta Q_{C_3}(n) = C_3 \cdot V_{e_3}(n)$$

$$\Rightarrow V_s = V_{e_3} \cdot \frac{C_3}{C_0} \cdot \frac{z^{-1}}{1-z^{-1}}$$

Eq. 1 - 30

N.B. : En inversant les phases ϕ_1 et ϕ_2 , nous obtenons une fonction de transfert similaire, mais décalée d'une demi-période $T_{ech}/2$ [Bil88] :

$$V_s = -V_{e_1} \cdot \frac{C_1}{C_0} \cdot \frac{z^{-1/2}}{1-z^{-1}} - V_{e_2} \cdot \frac{C_2}{C_0} + V_{e_3} \cdot \frac{C_3}{C_0} \cdot \frac{z^{-1/2}}{1-z^{-1}}$$

Eq. 1 - 31

Bien entendu si l'entrée est échantillonnée aux instants $n.T_{ech}$ et bloquée durant T_{ech} , il y a égalité entre les Eq. 1 - 29 et Eq. 1 - 31. Cette dernière remarque prend toute son importance si nous nous souvenons que la sortie d'un CNA (utilisé dans une modulation Sigma Delta) est maintenue constante durant T_{ech} . Nous pouvons donc « réduire » la structure de filtrage en remplaçant par exemple une masse commandée par la phase ϕ_1 par une entrée. Un tel exemple est donné ci-dessous (Figure 1 - 44).

Considérons le modulateur Sigma Delta du premier ordre décrit §III.3.1.1 et intégrons le filtre précédemment présenté (Figure 1 - 43) :

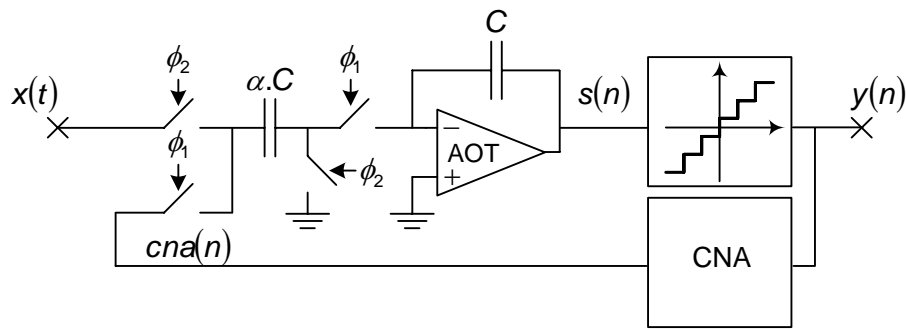


Figure 1 - 44 : Filtre à capacités commutées dans un modulateur Sigma Delta du premier ordre

Les opérations d'échantillonnage, de sommation/soustraction et de filtrage sont alors réalisées par une seule structure. De plus les filtres appliqués à l'entrée du modulateur $x(t)$ et à la sortie du CNA $cna(n)$ ne sont pas identiques (cf. Eq. 1 - 29 et la précédente remarque).

$$s(n) = \frac{\alpha}{1 - z^{-1}} \cdot (x(n) \cdot z^{-1} - cna(n)) \quad \text{Eq. 1 - 32}$$

Pour calculer la fonction de transfert du montage, nous remplaçons l'ensemble Quantificateur/CNA par un retard pur et un bruit blanc additif $e(n)$ et nous effectuons une transformée en z . L'expression de la fonction de transfert en z liée à ce type de montage est alors :

$$Y^*(z) = X^*(z) \cdot \underbrace{\frac{\alpha \cdot z^{-1}}{1 + (\alpha - 1) \cdot z^{-1}}}_{STF(z)} + E^*(z) \cdot \underbrace{\frac{1 - z^{-1}}{1 + (\alpha - 1) \cdot z^{-1}}}_{NTF(z)} \quad \text{Eq. 1 - 33}$$

IV.1.3. Structures avec rétroaction non capacitive

Il est également possible de substituer un système à capacité commutée à la capacité de rétroaction. Néanmoins il faut prendre toutes les précautions concernant l'amplificateur, afin d'éviter les

problèmes de stabilité lorsque, durant les transitions des phases ϕ_1 et ϕ_2 , la boucle de feedback est ouverte. Pour accroître la stabilité du montage une capacité en parallèle est alors ajoutée :

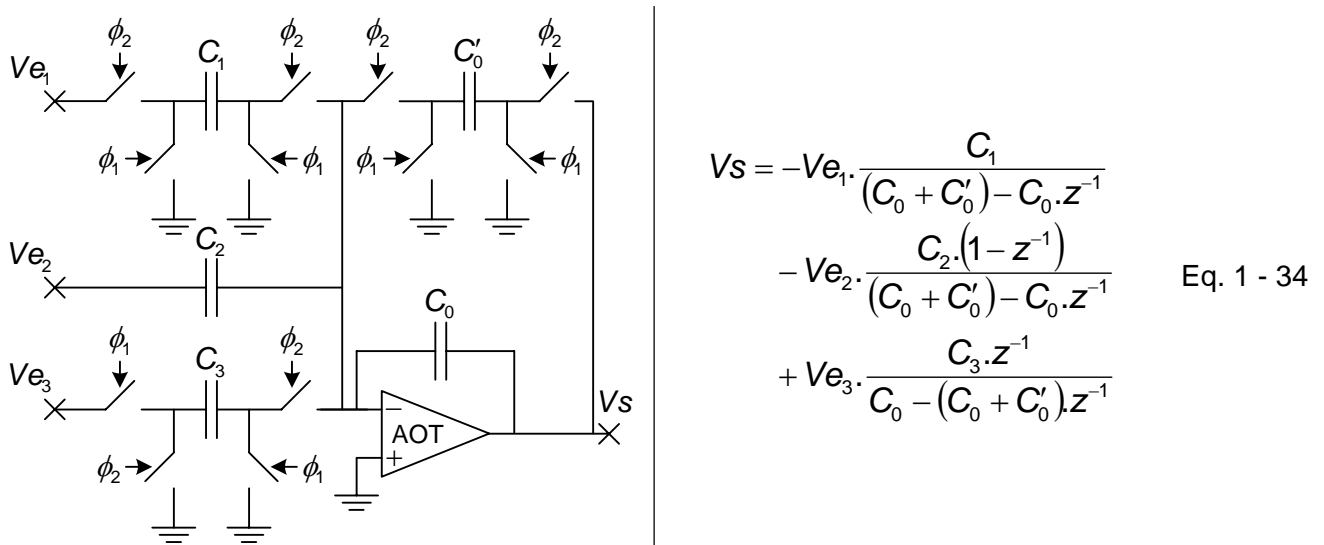


Figure 1 - 45 : Cellules de base avec capacité de rétroaction et structure à capacités commutées

Nous ne nous étendons pas d'avantage sur les exemples de circuits du premier ordre, les possibilités de réalisation d'autres circuits utilisant ces blocs élémentaires sont en théorie infinies. En pratique, des problèmes de stabilité empêchent un trop grand nombre de boucles.

De plus, nous avons considéré jusqu'à présent l'ensemble des éléments (interrupteur, capacité, Amplificateur Opérationnel à Transconductance) comme parfaits, ce qui n'est malheureusement pas conforme à la réalité. L'étude des conséquences sur le montage de ces imperfections sera l'objet la sous-partie suivante.

IV.2. Mise en œuvre des circuits à capacités commutées

Nous nous intéressons dans cette sous-partie principalement aux techniques mises en place pour diminuer ou annuler les effets des différentes imperfections des circuits à capacités commutées. En effet le but de cette sous-partie, plus proche de la technique, est non seulement d'appréhender les origines des bruits internes des structures à capacités commutées, mais également de nous rendre compte de la complexité de la mise en œuvre de ces structures. Cette complexité est telle, qu'elle peut engendrer des bruits qui lui sont propres. L'étude de ces bruits est au centre de la thèse.

Pour illustrer notre progression dans la complexité, nous utiliserons le circuit intégrateur suivant comme démonstrateur :

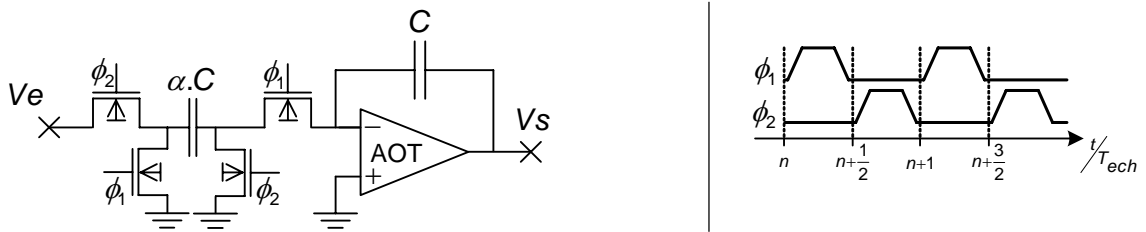


Figure 1 - 46 : Circuit intégrateur

IV.2.1. La non-linéarité de la résistance R_{ON} des interrupteurs MOS

Généralement, le premier problème auquel est confronté le concepteur concerne la linéarité (ou plutôt la non-linéarité) de la résistance de conduction R_{ON} des interrupteurs MOS (cf. Annexe 3 §1.3.1). Le problème de non-linéarité touche en priorité les interrupteurs soumis à de fortes excursions de tension, soit les deux situés à gauche dans la figure ci-dessous (les deux interrupteurs de droite sont reliés aux masses réelles et virtuelle de l'AOT). Nous restreindrons donc l'utilisation de la technique décrite dans l'Annexe 3 (cf. §1.3.2) à ces deux interrupteurs. La compensation de cette non-linéarité engendre alors le dédoublement des phases ϕ_1 et ϕ_2 en $(\phi_1; \bar{\phi}_1)$ et $(\phi_2; \bar{\phi}_2)$:

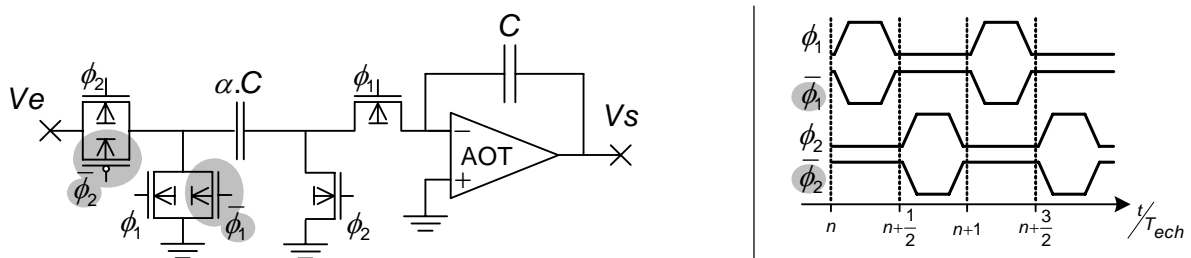


Figure 1 - 47 : Circuit intégrateur avec des interrupteur CMOS « linéaire »

La complexité du générateur d'horloge s'en trouve alors accrue.

IV.2.2. Les injections d'horloge des interrupteurs MOS

Le second problème auquel est confronté le concepteur concerne les charges parasites stockées à chaque cycle (charge/décharge) dans le canal de déplétion et les capacités parasites des interrupteur MOS (cf. Annexe 3 § 1.3.3). Ces charges faussent le transfert de charge entre les capacités et affectent ainsi le gain du montage. Pour contrer ce phénomène deux techniques sont généralement mises en œuvre. La première consiste à injecter une quantité de charges de signe opposé à celle stockée dans le NMOS (cf. Annexe 3 §1.3.4.1). Pour cela un PMOS de même taille est mis en parallèle avec le NMOS aux interrupteurs de droite et deux capacités MOS sont ajoutées aux interrupteurs de gauche afin de conserver leur linéarité. Cette technique est complétée par

une seconde technique qui consiste à jouer sur l'ordre d'ouverture et de fermeture des interrupteurs pour éviter la décharges des capacités parasites (cf. Annexe 3 §I.3.4.2).

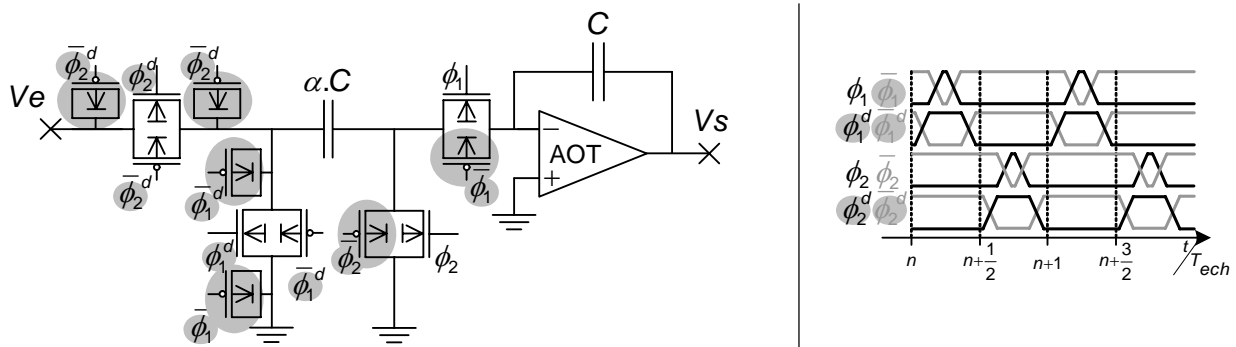


Figure 1 - 48 : Circuit intégrateur compensation des injection de charges

Nous constatons alors qu'outre la structure à capacité commutée, le générateur d'horloges de commande se complexifie également.

IV.2.3. Les capacités MOS

La capacité $\alpha.C$ est celle qui permet la mémorisation de la valeur du signal d'entrée. Elle est donc plus sensible d'un point de vue de ces propres capacités parasites (cf. Annexe 3 §I.3.5). Il n'existe pas à proprement parler de technique permettant d'annuler les effets de ces capacités parasites. Il est cependant intéressant de placer la plus grosse capacité parasite en entrée comme cela est décrit ci-dessous :

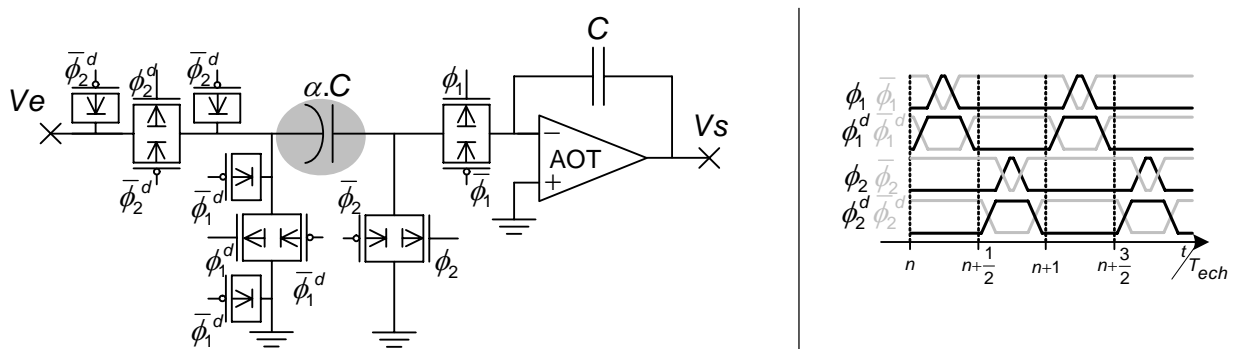


Figure 1 - 49 : Schéma complet du circuit intégrateur

Une fois toutes ces améliorations apportées, nous obtenons donc un circuit plus complexe aussi bien concernant les transistors mais également, et c'est ce qui nous intéressera par la suite, concernant les chronogrammes de commande qui nécessitent des générateurs d'horloges assez complexe.

IV.3. Circuits générateurs d'horloges

Comme nous l'avons vu précédemment, la chronologie d'ouverture et de fermeture des interrupteurs MOS revêt une grande importance, tant dans la définition même de la fonction de transfert du filtre, que dans l'annulation ou la minimisation des défauts dus aux capacités parasites.

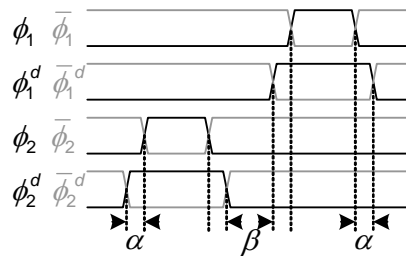


Figure 1 - 50 : Chronogramme de commande des interrupteurs CMOS

A l'image des chronogrammes, les générateurs de signaux deviennent plus complexes. Ils sont en général composés de cellules de retard (inverseurs de composition différente afin de créer plus ou moins de retard) bouclées en Latch RS qu'il faut ajuster afin d'obtenir les largeurs et les différents décalages (α, β) entre les divers signaux d'horloge :

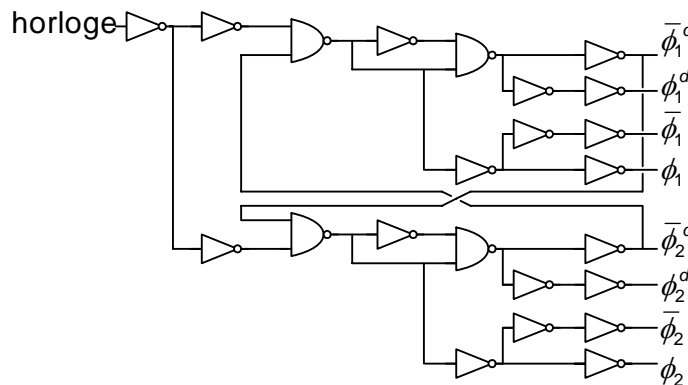


Figure 1 - 51 : Générateur d'horloges non-recouvrantes

Comme cela est décrit ci-dessus, ce type de générateur d'horloges est un système bouclé ainsi la moindre défaillance même temporaire d'une ou de plusieurs de ses composantes a des répercussions sur l'ensemble des horloges générées. L'étude de l'impacte des variations temporelles de ces phases d'horloge (appelées jitter d'horloge) sur le modulateur Sigma Delta est l'objet de cette thèse.

V. CONCLUSION

Les bases techniques nécessaires à la réalisation de nos travaux sont maintenant posées. L'étude, la caractérisation et la modélisation des différentes sources de bruit vont pouvoir débiter. Avant de nous lancer dans nos travaux il faut nous poser quelques questions inhérentes à l'étude que nous venons de réaliser :

Quel type de bruit allons nous rencontrer ? L'étude globale des chaînes de réception RF, conduite en partant des différentes architectures de réception existantes pour finir sur l'architecture retenue par la société, a mis en évidence la multiplicité des blocs composants la chaîne de réception RF. Ce sont autant de sources et de types de bruits potentiels. L'étude ainsi que la modélisation des différents types de bruits et de leurs effets sur les structures seront l'objet du chapitre suivant (cf. Chapitre II).

Quelle est l'influence du jitter sur une structure simple à capacités commutées ? En effet le choix de l'architecture de réception RF a également conditionné le choix du type de convertisseur analogique/numérique. Les structures de conversion classiques ne répondant pas au cahier des charges, les concepteurs se sont tournés vers des modulateurs Sigma Delta passe-bande réels et complexes. L'utilisation de modulateurs à temps discret a induit l'utilisation de filtre ayant une architecture à capacités commutées. Et, pour finir, l'étude des différentes méthodes de conception de ces structures a mis en évidence une grande complexité des structures de commande des interrupteurs CMOS. Cette complexité va inévitablement entraîner une multiplication des sources de jitter multipliant ainsi leur effet sur les architectures à capacités commutées. L'étude de l'influence du jitter sur une structure simple sera l'objet du Chapitre III.

Quelle est la part du jitter sur les performances globales de la chaîne de réception RF ? En effet, comme nous venons de l'étudier, la chaîne de réception RF forme un tout difficilement dissociable où de multiples sources de bruit se combinent affectant les performances finales de la structure. Cette dernière question sera traitée dans le Chapitre IV.

CHAPITRE II :

Modélisation et caractérisation de
sources de bruits en temps discret

I. INTRODUCTION

Les chaînes de réception RF décrites dans le chapitre précédent sont des structures comportant de nombreux blocs parfois de grande complexité. Les performances de ces architectures se mesurent en général dans le domaine fréquentiel. Des simulateurs tels que SpectreRF [Spe03] ou EldoRF [Eld04] sont alors utilisés pour les évaluer. Les simulations fréquentielles (AC, Noise, PSS, SST, etc...) au niveau des transistors permettent de vérifier le bon fonctionnement ou d'estimer le bruit ou les non-linéarités. Cependant, ce type de simulations reste cantonné à un ou deux blocs, car les moyens, logiciels et machines, dont nous disposons ne permettent pas la prise en compte des performances globales de la chaîne. Nous aurons donc recours à la modélisation comportementale pour vérifier le comportement de la chaîne complète.

Il existe une catégorie de modèles comportementaux, appelés « macro-modèles » constitués d'éléments idéaux : sources de courant/tension, résistances, capacités, interrupteurs, etc... Ces modèles présentent l'avantage d'être simples à mettre en place et à utiliser. Ils sont très utiles lors des phases de conception des blocs et permettent d'effectuer de rapides simulations pour vérifier le bon fonctionnement d'une brique de conception dans un ensemble. Cependant, leur complexité augmente dès que l'on souhaite intégrer des imperfections telles que le bruit ou les non-linéarités, ce qui les rend impropres aux simulations de bruit ou de non-linéarités sur l'ensemble de la chaîne.

Lorsque la complexité du montage est trop grande, les concepteurs ont alors recours aux langages de modélisation comportementale comme le Verilog-AMS [VER], [EDA] ou le VHDL-AMS [IEE99] [Her02]. Ces langages permettent l'écriture de modèles sous forme d'équations mathématiques tout en respectant les lois physiques qui régissent l'électronique. Ils disposent d'instructions pour modéliser le bruit mais les simulations restent dans le domaine fréquentiel (petits signaux). Ces modèles ne conviennent pas aux simulations que nous voulons effectuer, car les simulations linéaires fréquentielles ne sont pas adaptées aux circuits mixtes, notre domaine d'application.

Dans ce contexte, nous avons choisi de modéliser et simuler les bruits dans le domaine temporel et d'effectuer ensuite un traitement fréquentiel post-simulation approprié. Nous utilisons pour cela le VHDL-AMS [IEE99] [Her02] et un simulateur associé ADVanceMS [ADV00]. Ils permettent une description fonctionnelle temporelle dite « pilotée par événements » particulièrement appropriée aux structures à capacités commutées ou aux circuits mixtes en général et ce grâce à la présence de

deux noyaux de simulation synchronisés : l'un évènementiel (Modelsim) et l'autre temps continu (Eldo).

Ainsi nous avons donc entrepris de constituer une bibliothèque de modèles VHDL-AMS permettant la prise en compte du bruit au niveau comportemental.

Pour mener à bien notre travail de modélisation, une connaissance approfondie des différents bruits présents dans les circuits électroniques et de leurs caractéristiques temporelles et fréquentielles est indispensable. De même, l'étude de l'influence de ces bruits « intrinsèques » sur des circuits à temps discrets et sur des générateurs (de sinusoïdes ou de carrés) doit être menée. Ces études seront valables quel que soit le simulateur, ainsi elles pourront être réutilisées pour implanter des modèles dans d'autres langages de modélisation.

Ce chapitre est structuré en trois parties. La première concerne l'étude théorique des sources de bruit dont découlent les relations liant les domaines temporel et fréquentiel. La deuxième concerne la modélisation de ces bruits. Leur intégration dans des générateurs de signaux sinusoïdaux et carrés fera l'objet de la troisième partie. Ainsi une étude ultérieure pourra être menée concernant l'influence du bruit de phase ou de jitter des générateurs externes pilotant un modulateur Sigma Delta : le signal sinus et l'horloge.

II. ETUDE THEORIQUE DES BRUITS

II.1. Rappels

II.1.1. Calcul de la puissance

Nous pouvons dissocier deux types de signaux, ceux à énergie finie et ceux à puissance finie. Nous nous intéressons, dans ce chapitre, aux signaux à énergie infinie et à puissance finie. Cela se traduit par une puissance moyenne constante. Les signaux sinusoïdaux ou les bruits font partie de cette catégorie. La puissance moyenne P_x d'un signal $x(t)$ et sa valeur efficace X sont définies par :

$$P_x = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} |x(t)|^2 dt \quad \text{et} \quad X = \sqrt{P_x} \quad \text{Eq. 2 - 1}$$

Cependant, autant le calcul de la puissance par cette formule est aisé pour un signal déterministe (dont la fonction mathématique est connue, une sinusoïde, par exemple), autant l'utilisation de cette formule est impossible pour des phénomènes ayant une évolution aléatoire dans le temps, comme c'est le cas pour le bruit. La description d'un signal aléatoire $b(t)$, se fait alors à travers les moments d'ordre k notés $m_{k,b}$ et définis par :

$$m_{k,b} = E[b^k] = \int_{-\infty}^{+\infty} u^k \cdot p_b(u) du \quad \text{Eq. 2 - 2}$$

$E(.)$ désigne l'espérance mathématique et $p_b(.)$ la densité de probabilité de $b(t)$.

En nous plaçant dans le cas de signaux ergodiques, pour lesquels les moyennes temporelles sont égales aux moyennes statistiques (ou moment), nous pouvons utiliser l'égalité suivante :

$$m_{2,b} = \int_{-\infty}^{+\infty} u^2 \cdot p_b(u) du = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} |b(t)|^2 dt = P_b \quad \text{Eq. 2 - 3}$$

P_b désigne la puissance moyenne du phénomène aléatoire $b(t)$.

Pour le calcul de la Densité Spectrale de Puissance (DSP) d'un bruit, nous utilisons la relation de PARSEVAL [DEQ96] :

$$P_b = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} |b(t)|^2 \cdot dt = \int_{-\infty}^{+\infty} S_b(f) \cdot df \quad \text{Eq. 2 - 4}$$

Où $S_b(f)$ désigne la fonction de DSP du bruit $b(t)$.

Comme nous bornons notre étude aux signaux discrets, nous pouvons introduire une nouvelle restriction sur le domaine d'intégration. Nous notons F_{ech} la fréquence d'échantillonnage du montage, nous utiliserons cette même fréquence pour l'analyse fréquentielle (Transformée de Fourier). En utilisant la propriété de périodicité du spectre, nous pouvons alors réduire l'étude fréquentielle à l'intervalle $[-F_{ech}/2, F_{ech}/2]$, d'où :

$$P_b = \int_{-F_{ech}/2}^{+F_{ech}/2} S_b(f).df \quad \text{Eq. 2 - 5}$$

Comme l'évaluation de la fonction de DSP du bruit ($S_b(f)$) ne peut être effectuée de manière continue par nos outils de simulation ou de mesure, nous utilisons une approximation discrétisée de cette fonction. Cette approximation est obtenue en divisant l'intervalle d'étude en bandes de fréquences Δf , centrées autour des fréquences f sur lesquelles la DSP $S_b(f)$ est considérée comme constante.

C'est en fait une puissance $P(f)_{\Delta f}$ exprimée en V_{rms}^2 calculée sur une bande de fréquence Δf centrée autour de la fréquence f :

$$S_b(f).\Delta f \cong P(f)_{\Delta f} = \int_{f-\Delta f/2}^{f+\Delta f/2} S_b(u).du \quad \text{en } V_{rms}^2 \quad \text{Eq. 2 - 6}$$

N.B. : Si l'intervalle de fréquences Δf est égal à 1Hz alors $S_b(f) \cong P(f)_{\Delta f}$. Nous retrouvons bien le fait que la DSP correspond à une puissance par unité de fréquence.

II.1.2. Unités de mesure de la Densité Spectrale de Puissance

Si le bruit $b(t)$ est mesuré en Volts (bruit de tension), alors l'unité la plus immédiate pour $S_b(f)$ est le V_{rms}^2/Hz . Le V_{rms}/\sqrt{Hz} est utilisé comme une unité se rapportant à la valeur efficace $\sqrt{S_b(f)}$ (cf. Eq. 2 - 1). La DSP peut également s'exprimer en décibels : $S_b(f)_{dB_v/Hz} = 10.\log(S_b(f)_{V_{rms}^2/Hz})$.

Cependant, lors de mesures ou de simulations, il est parfois difficile de déterminer l'origine ou la puissance référence ($0 \text{ dB}_v/Hz = 10.\log(1 V_{rms}^2/Hz)$). C'est notamment le cas lorsque l'on utilise la Transformée de Fourier non normalisée. On utilise, dans ce cas, une puissance connue qui sert d'étalon. On définit alors des unités par comparaison.

Nous utiliserons comme puissance étalon :

- soit la puissance connue d'une porteuse, l'unité sera alors le dB_c / Hz (le c en indice vient de l'anglais carrier, pour porteuse)
- soit une puissance d' $1mW$ dissipée dans une résistance de 50Ω , l'unité sera alors le $dB_{m50\Omega} / Hz$ (généralement noté dB_m / Hz)

II.1.2.1. Le dB_c / Hz

Cette unité est principalement utilisée pour évaluer le bruit de phase d'un oscillateur. Elle représente le rapport entre la DSP du signal étudié (bruité en l'occurrence) sur une bande de fréquence de $1Hz$ aux fréquences $f = f_o \pm f_m$ (car le spectre est généralement symétrique par rapport à la porteuse) et la puissance d'une porteuse de référence de fréquence f_o . f_m désigne la fréquence d'offset par rapport à la fréquence de la porteuse f_o . La DSP de bruit en dB_c / Hz notée $S_b(f)_{dB_c / Hz}$ est égale à :

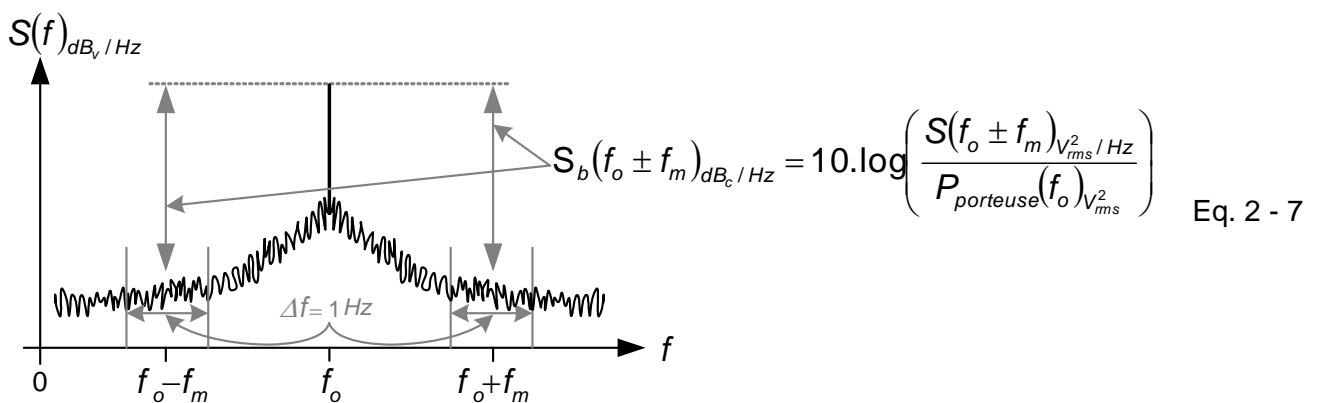


Figure 2 - 1 : Définition de la DSP de bruit en dB_c / Hz

N.B. : Dans la représentation ci-dessus, la puissance des raies de fréquences négatives a été repliée sur les fréquences positives, afin de respecter PARSEVAL.

Comme la mesure de $S_b(f)_{V_{rms}^2 / Hz}$ en haute fréquence est parfois difficile notamment lorsqu'on mesure le bruit de phase d'une porteuse à la fréquence f_o , on a alors recours à une extension de cette formule : $L(f_m)$. Cette mesure consiste à ramener autour de la fréquence nulle (souvent par le jeu de mixers) la puissance auparavant dissipée autour de la porteuse. Cette opération engendre alors la représentation décrite par la Figure 2 - 2.

La puissance de bruit des fréquences négatives ($f_m < 0$) étant ramenée sur un seul côté de la porteuse ($f_m > 0$), la DSP représentée est alors doublée. A partir de Eq. 2 - 7 et Eq. 2 - 8, on obtient une relation entre $L(f)_{dB_c / Hz}$ et $S_b(f)_{dB_c / Hz}$ [HaL99] :

$$L(f_m)_{dB_c / Hz} = 10.\log\left(\frac{S'_b(f_m)_{V_{rms}^2 / Hz}}{P_{porteuse}(f_c)_{V_{rms}^2}}\right) = 10.\log\left(\frac{2.S_b(f_c \pm f_m)_{V_{rms}^2 / Hz}}{P_{porteuse}(f_c)_{V_{rms}^2}}\right) \quad \text{Eq. 2 - 8}$$

$$L(f_m)_{dB_c / Hz} = 10.\log(2) + S_b(f_c \pm f_m)_{dB_c / Hz}$$

N.B. : $L(f_m)$ est principalement utilisée par les bancs de mesure de bruit de phase. Nous n'y aurons pas recours pour la représentation des bruits modélisés.

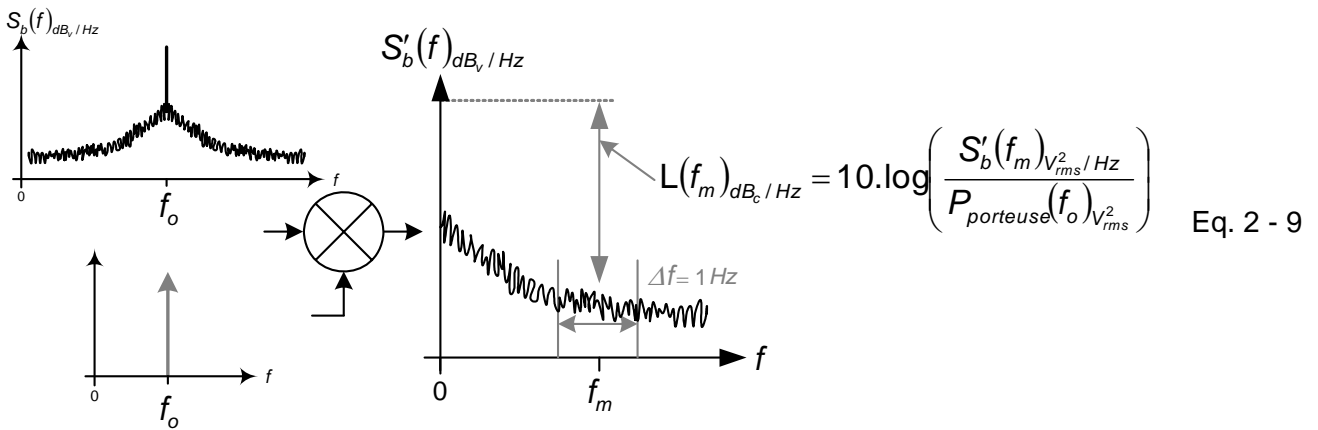


Figure 2 - 2 : Définition de $L(f_m)$ en dB_c / Hz

II.1.2.2. Le $dB_{m50\Omega} / Hz$

Cette unité ($dB_{m50\Omega} / Hz$ notée par la suite : dB_m / Hz) est issue du domaine hyperfréquence pour des circuits d'impédance caractéristique 50Ω . Elle représente le rapport entre la DSP du bruit sur une bande de $1Hz$ à la fréquence f et une puissance de $1mW$ toutes les deux dissipées dans une résistance de 50Ω .

Ainsi la DSP en dB_m / Hz notée $S_b(f)_{dB_m / Hz}$ s'exprime :

$$S_b(f)_{dB_m / Hz} = 10.\log\left(\frac{S_b(f)_{V_{rms}^2 / Hz} \text{ dans } 50\Omega}{1mW}\right)$$

$$= 10.\log\left(\frac{(V_b(\sqrt{f})_{V_{rms} / \sqrt{Hz}})^2 / 50\Omega}{1mW}\right) \quad \text{Eq. 2 - 10}$$

$$= 10.\log\left((V_b(\sqrt{f})_{V_{rms} / \sqrt{Hz}})^2\right) - 10.\log(10^{-3} \times 50)$$

$$\approx 20.\log(V_b(\sqrt{f})_{V_{rms} / \sqrt{Hz}}) + 13.01dB_v$$

La figure suivante illustre le calcul de $S_b(f)_{dB_m / Hz}$:

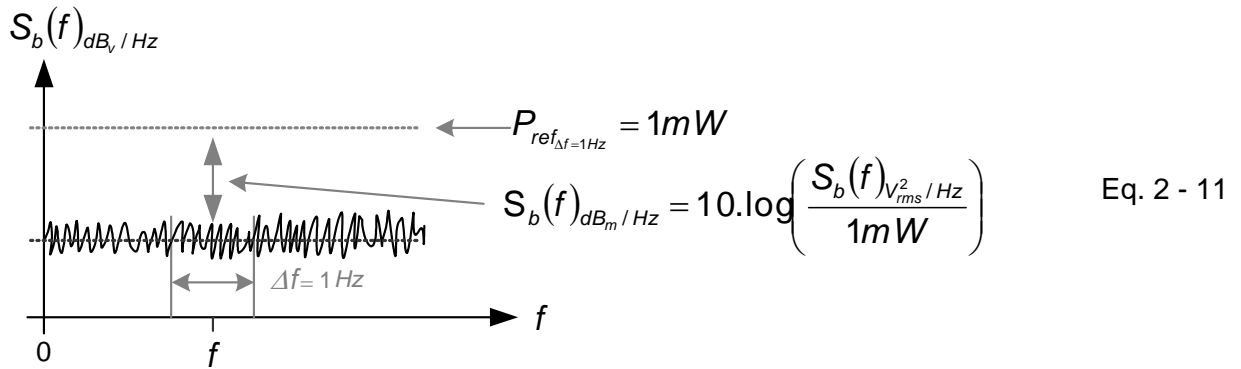


Figure 2 - 3 : Définition de la DSP de bruit en dB_m / Hz

II.1.3. Analyse spectrale

Pour le traitement fréquentiel des données, nous avons eu recours à des algorithmes de FFT (Fast Fourier Transform) et de périodogramme [StM97] développées sous SCILAB [SCI] par M. ROBBE et fondés sur les travaux suivants : [Wei67], [Har78] et [Nut81].

Compte tenu de la nature aléatoire des signaux analysés, le traitement fréquentiel choisi est un périodogramme moyenné et normalisé par rapport à $0 dB_v = 10.Log(1 V_{rms}^2)$, c'est-à-dire que toutes les valeurs obtenues sont directement en dB_v (§ II.1.2). Comme nous effectuons une acquisition sur un nombre d'échantillons fini $N = 2^m$, la DSP du bruit est discrétisée sur une bande de fréquence minimale $\Delta f = F_{ech}/N$. La figure suivante illustre cette discrétisation de la DSP du bruit :

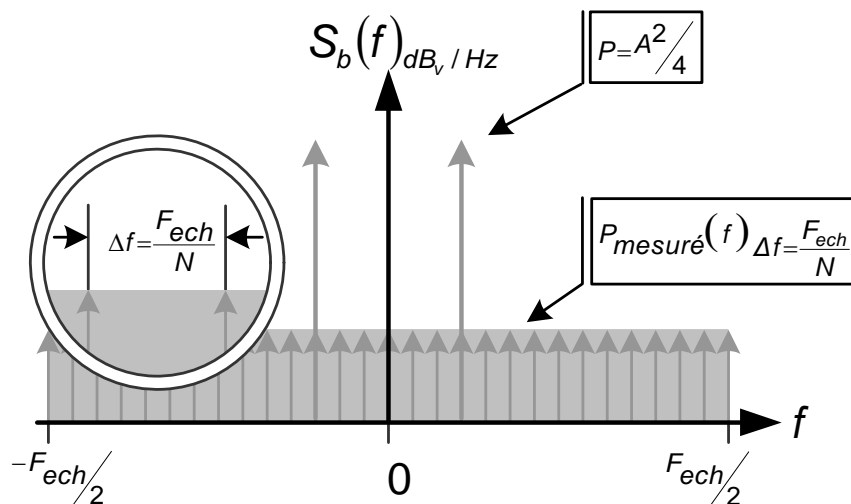


Figure 2 - 4 : Traitement du spectre sous SCILAB

N.B. : La DSP d'une porteuse sinusoïdale d'amplitude A est concentrée sur deux raies fréquentielles de hauteur invariable $A^2/4$.

En ce qui concerne le niveau de bruit, pour obtenir une densité spectrale de puissance ramenée à une bande de 1Hz , il nous faudra appliquer la transformation suivante :

$$S_{théorique}(f)_{V_{rms}^2/Hz} = P_{théorique}(f)_{\Delta f=1Hz} = \frac{P_{mesuré}(f)_{\Delta f=F_{ech}/N}}{F_{ech}/N} \quad \text{Eq. 2 - 12}$$

En tenant compte de l'équation précédente la DSP de bruit en dB/Hz est alors égale à :

$$S_{théorique}(f)_{\text{dB}/\text{Hz}} = S_{mesuré}(f)_{\text{dB}/\Delta f} - 10 \cdot \log\left(\frac{F_{ech}}{N}\right) \quad \text{Eq. 2 - 13}$$

Dans notre cas, nous appliquons, en plus, un fenêtrage pour le calcul du spectre qui permet d'éviter les phénomènes d'étalement d'énergie. Le facteur F_{ech}/N est alors remplacé par le facteur BBE (Bande de Bruit Equivalente) [Har78], calculé en interne par le logiciel et qui tient compte du nombre de points N , de la fréquence d'échantillonnage F_{ech} et du fenêtrage. Ainsi l'équation précédente devient :

$$S_{théorique}(f)_{\text{dB}/\text{Hz}} = S_{mesuré}(f)_{\text{dB}/\Delta f} - 10 \cdot \log(\text{BBE}) \quad \text{Eq. 2 - 14}$$

II.2. Le bruit au niveau des composants électroniques

Nous distinguerons deux types de bruits internes affectant un composant électronique :

- les bruits blancs c'est-à-dire de DSP constante quelle que soit la fréquence d'étude comme le bruit thermique et le bruit de grenaille (shot noise) ;
- les bruits roses dont la DSP décroît lorsque la fréquence d'étude croît, comme le bruit de Flicker ($1/f$) et le bruit de popcorn ($1/f^2$) qui ne sera pas étudié dans ce document car sa contribution est généralement considérée comme négligeable.

Le bruit total d'un composant est généralement la combinaison de plusieurs de ces bruits.

II.2.1. Le bruit thermique

L'identification et la caractérisation du bruit thermique sont le fruit des travaux de Johnson [Joh28] et Nyquist [Nyq28]. Ce bruit est dû aux mouvements aléatoires des porteurs de charges dans un conducteur. Il affecte en particulier les résistances mais également le courant de drain et de grille pour les transistors MOS (cf. Figure 2 - 8) [Lee98].

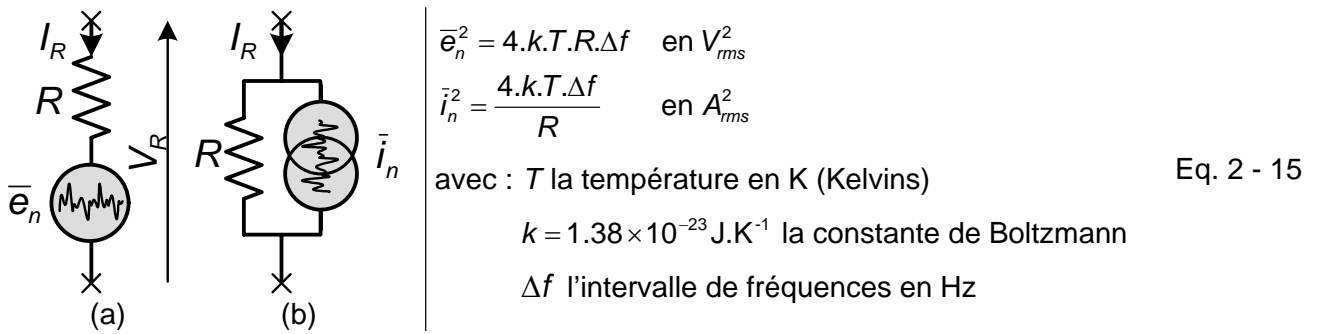


Figure 2 - 5 : Bruit thermique dans une résistance

N.B. : la puissance du bruit dans une résistance de 50 Ohms à température ambiante (300°K) est alors : $\bar{e}_n = 0.9 \text{ nV}_{rms} / \sqrt{\text{Hz}}$

II.2.2. Le bruit de grenaille

Ce bruit a été décrit et caractérisé pour la première fois par Schottky [Sch18]. Ce bruit est dû au saut des barrières de potentiel par les porteurs de charges. Il est donc situé au niveau des jonctions traversées par un courant. Il affecte principalement les diodes et les transistors bipolaires (cf. Figure 2 - 7). Sa DSP vaut [Lee98] :

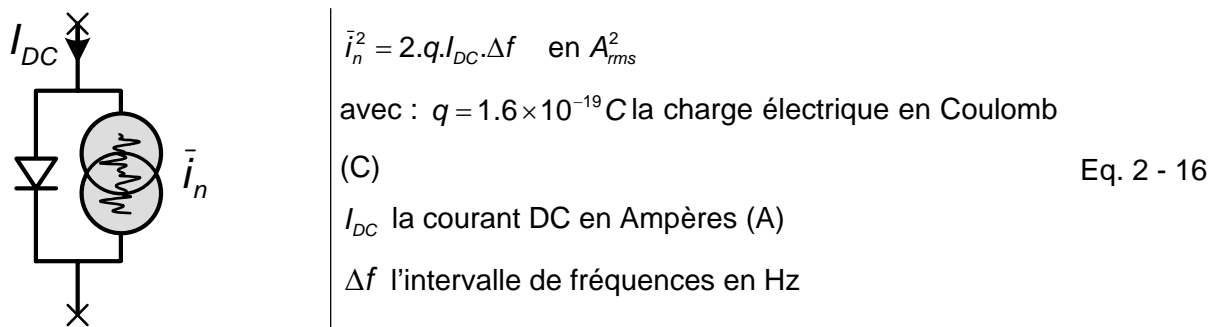
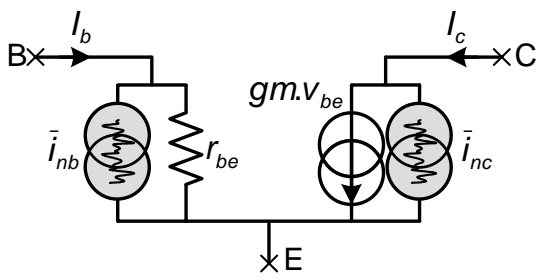


Figure 2 - 6 : Bruit de grenaille dans une diode

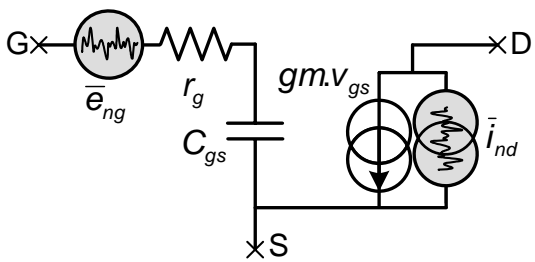
II.2.3. Le bruit de Flicker

Ce bruit est également appelé bruit en $1/f$ en référence à la variation de sa DSP en $1/f$. Actuellement aucun mécanisme universel ne permet d'expliquer ce bruit. Il affecte les résistances, les transistors MOS et le courant de collecteur/émetteur des transistors bipolaires où il peut être expliqué par des phénomènes de combinaison/recombinaison des porteurs de charges. Nous développons ci-dessous deux modélisations des bruits dans un transistor bipolaire et un transistor MOS [Lee98] :



$$\begin{aligned} \bar{i}_{nb}^2 &= 2 \cdot q \cdot I_B \cdot \Delta f + \frac{K_1}{f} \cdot \Delta f && \text{en } A_{rms}^2 \\ \bar{i}_{nc}^2 &= 2 \cdot q \cdot I_C \cdot \Delta f && \text{en } A_{rms}^2 \end{aligned} \quad \text{Eq. 2 - 17}$$

Figure 2 - 7 : Bruits dans un transistor bipolaire



$$\begin{aligned} \bar{i}_{nd}^2 &= 4 \cdot k \cdot T \cdot \gamma \cdot \frac{1}{5 \cdot r_g} \cdot \Delta f + \frac{K}{f} \cdot \Delta f && \text{en } A_{rms}^2 \text{ [Zie62]} \\ \bar{e}_{ng}^2 &= 4 \cdot k \cdot T \cdot \delta \cdot \frac{1}{5 \cdot r_g} \cdot \Delta f && \text{en } V_{rms}^2 \text{ [Zie86]} \end{aligned} \quad \text{Eq. 2 - 18}$$

Figure 2 - 8 : Bruits dans un transistor MOS

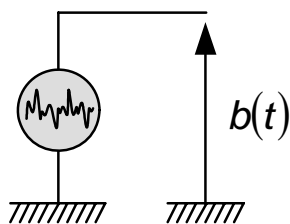
Toutes ces sources de bruits se combinent au sein d'un circuit pour générer du bruit d'amplitude, de phase, de jitter. Examinons différents cas susceptibles d'être rencontrés dans les circuits à temps discret.

II.3. Manifestation du bruit dans les circuits à temps discret

Dans cette partie, nous considérons l'ensemble des bruits comme des variations aléatoires cadencées à la fréquence F_{ech} qui sera également notre fréquence d'échantillonnage.

II.3.1. Bruit blanc de tension

Nous considérons comme illustrée ci-dessous une source de bruit blanc de tension $b(t)$:



Le bruit $b(t)$ suit la loi normale $N(0, \sigma)$ de moyenne nulle et d'écart type égal à σ (exprimé en V_{rms}).

Figure 2 - 9 : Source de bruit blanc de tension

Sachant qu'un bruit blanc $b(t)$, de type gaussien $N(0, \sigma)$, possède une DSP $S_b(f)$ constante quelle que soit la fréquence et a une puissance égale à σ^2 (σ^2 désigne également la variance de la loi normale), nous pouvons écrire à l'aide de Eq. 2 - 5 :

$$P_b = \sigma^2 = \int_{-F_{ech}/2}^{+F_{ech}/2} S_b(f).df \quad \text{Eq. 2 - 19}$$

$$= S_b(f).F_{ech} \quad \text{en } V_{rms}^2$$

La DSP s'écrit alors :

$$S_b(f)_{V_{rms}^2/Hz} = \sigma^2 / F_{ech} \quad \text{Eq. 2 - 20}$$

L'expression de cette DSP en dB_v/Hz devient alors :

$$S_b(f)_{dB_v/Hz} = 10.\log\left(\frac{\sigma^2}{F_{ech}}\right) \quad \text{Eq. 2 - 21}$$

Soit en dB_m/Hz (cf. Eq. 2 - 10) :

$$S_b(f)_{dB_m/Hz} \approx 10.\log\left(\frac{\sigma^2}{F_{ech}}\right) + 13.01dB_v \quad \text{Eq. 2 - 22}$$

II.3.2. Bruit d'amplitude d'une sinusoïde

Nous venons d'étudier le bruit blanc de tension, intéressons nous maintenant au bruit d'amplitude d'une sinusoïde. Ce bruit peut provenir, par exemple, de l'addition d'un bruit blanc sur le gain d'un transistor MOS ou d'un amplificateur. Considérons une source sinusoïdale $y(t)$ sujette à une variation relative $b_A(t)$ de son amplitude A :

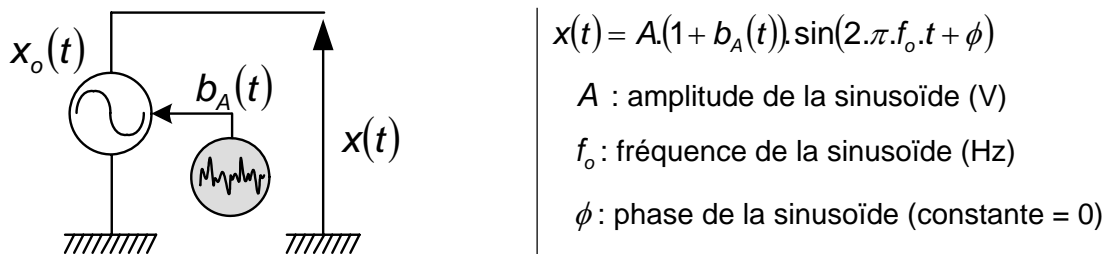


Figure 2 - 10 : Bruit d'amplitude d'une sinusoïde

N.B. : Nous considérons que le bruit $b_A(t)$ (signal aléatoire) et la porteuse (signal déterministe) sont indépendants.

Le bruit blanc $b_A(t)$ suit la loi normale $N(0, \sigma_A)$ de moyenne nulle et d'écart type égal à σ_A . La DSP de ce bruit en dB_V/Hz sera notée $S_{b_A}(f)_{dB_V/Hz}$.

$x(t)$ peut également se décomposer de la façon suivante :

$$x(t) = \underbrace{A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t + \phi)}_{\text{porteuse idéale } x_o(t)} + \underbrace{A \cdot b_A(t) \cdot \sin(2 \cdot \pi \cdot f_o \cdot t + \phi)}_{\text{bruit d'amplitude additif}} \quad \text{Eq. 2 - 23}$$

Nous effectuons maintenant un bilan de puissance des différentes composantes. La puissance P_{x_o} de la porteuse idéale $x_o(t) = A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t)$ est égale à (cf. Eq. 2 - 1) :

$$P_x(f_o)_{V_{rms}^2} = A^2 / 2 \quad \text{Eq. 2 - 24}$$

Pour le calcul de la DSP du bruit d'amplitude, nous utiliser la propriété de l'espérance mathématique $E(.)$ suivante :

$$E(X^2 \cdot Y^2) = E(X^2) \cdot E(Y^2) \quad \text{Eq. 2 - 25}$$

Ainsi (cf. Eq. 2 - 3) la puissance du bruit d'amplitude, notée $P_{b.amp}$, s'exprime :

$$\begin{aligned} P_{b.amp} &= E\left(\left|b_A(t) \cdot A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t)\right|^2\right) \\ &= \underbrace{E\left(\left|b_A(t)\right|^2\right)}_{P_{b_A}} \cdot \underbrace{E\left(\left|A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t)\right|^2\right)}_{P_x} \\ &= \sigma_A^2 \cdot A^2 / 2 \quad \text{en } V_{rms}^2 \end{aligned} \quad \text{Eq. 2 - 26}$$

La DSP du bruit additif d'amplitude s'écrit alors :

$$S_{amp}(f)_{V_{rms}^2/Hz} = (\sigma_A \cdot A)^2 / 2 \cdot F_{ech} \quad \text{Eq. 2 - 27}$$

La DSP de la porteuse d'amplitude bruitée en dB_C/Hz , notée $S_x(f)_{dB_C/Hz}$, est égale à :

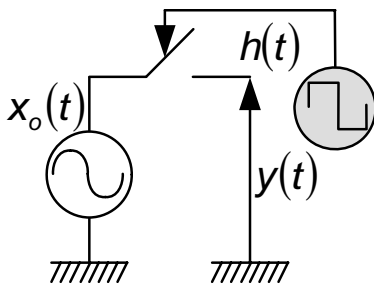
$$\begin{aligned} S_x(f)_{dB_C/Hz} &= 10 \cdot \log\left(\frac{S_{amp}(f)_{V_{rms}^2/Hz}}{P_{x_o}(f_o)_{V_{rms}^2}}\right) \\ &= 10 \cdot \log\left(\frac{\sigma_A^2}{F_{ech}}\right) \\ &= S_{b_A}(f)_{dB_V/Hz} \end{aligned} \quad \text{Eq. 2 - 28}$$

N.B. : On obtient donc, dans le domaine fréquentiel, un bruit d'amplitude centré autour de la porteuse et dont la DSP en dB_C/Hz est égale à la DSP du bruit blanc en dB_V/Hz (nous verrons § II.3.4 que cela n'est pas totalement exact).

II.3.3. Bruit blanc d'échantillonnage

Lorsque la source de bruit blanc vient s'ajouter à la tension de seuil d'une bascule ou d'un interrupteur MOS, elle entraîne une incertitude sur l'instant d'action (transfert de données, échantillonnage, etc...). Cette incertitude temporelle est appelée jitter. Lorsqu'un tel jitter affecte un générateur piloté par une horloge, comme un diviseur de fréquence ou un séquenceur utilisé notamment dans les architectures à capacités commutées, ce jitter porte le nom de jitter PM (Phase Modulation) [Kun98].

Comme la mise en évidence directe du jitter n'est pas explicite, nous utilisons l'échantillonnage d'une porteuse idéale $x_o(t)$. Nous considérons alors un échantillonneur bloqueur commandé par une horloge $h(t)$ sujette au jitter PM. Dans le cas idéal, l'instant d'échantillonnage est défini par $t_n = n.T_{ech}$ et par $t'_n = T_{ech}(n + b_{PM}(n))$ dans le cas bruité. $T_{ech} \cdot b_{PM}(n)$ désigne la perturbation temporelle de l'instant d'échantillonnage.



$$\begin{aligned}
 y_{ideale}(t) &= \sum_{n=-\infty}^{\infty} x_o(t_n) \delta(t - t_n) \\
 &= \sum_{n=0}^{\infty} A \sin(2\pi \cdot f_o \cdot n T_{ech}) \delta(t - n T_{ech}) \\
 y_{bruitée}(t) &= \sum_{n=0}^{\infty} x_o(t'_n) \delta(t - t'_n) \\
 &= \sum_{n=0}^{\infty} A \sin(2\pi \cdot f_o \cdot T_{ech}(n + b_{PM}(n))) \delta(t - T_{ech}(n + b_{PM}(n)))
 \end{aligned}
 \tag{Eq. 2 - 29}$$

Figure 2 - 11 : Echantillonnage idéal et bruité d'une porteuse

$y_{ideale}(t)$ est la sinusoïde échantillonnée idéalement et $y_{bruitée}(t)$ représente ce même signal mais échantillonné par une horloge bruitée. $b_{PM}(n)$ suit la loi normale $N(0, \sigma_{PM})$ avec une valeur moyenne nulle et un écart-type égal à σ_{PM} . La DSP de bruit en dB_V/Hz sera notée : $S_{b_{PM}}(f)_{dB_V/Hz}$. $\delta(t)$ désigne l'impulsion de DIRAC.

Comme développé ci-dessous, nous pouvons décomposer le signal $x_o(t'_n)$ en la somme d'un signal idéalement échantillonné et d'un bruit proportionnel à la dérivée de ce signal [BoW88] :

$$\begin{aligned}
 x_o(t'_n) &= A \cdot \sin(2\pi \cdot f_o \cdot T_{ech} \cdot (n + b_{PM}(n))) \quad \text{avec } 2\pi \cdot f_o \cdot T_{ech} \cdot b_{PM}(n) \ll 1 \\
 &\approx A \cdot (\sin(2\pi \cdot f_o \cdot n T_{ech}) + 2\pi \cdot f_o \cdot T_{ech} \cdot b_{PM}(n) \cdot \cos(2\pi \cdot f_o \cdot n T_{ech})) \\
 &\approx x_o(t_n) + \frac{\partial x_o}{\partial t}(t_n) \cdot T_{ech} \cdot b_{PM}(n)
 \end{aligned}
 \tag{Eq. 2 - 30}$$

Pour analyser ce type de comportements, nous avons recours à une FFT. Notons que, comme illustré par la figure suivante, l'acquisition des échantillons pour la FFT s'effectuera à l'instant idéal t_n décalé d'une demi-période d'échantillonnage :

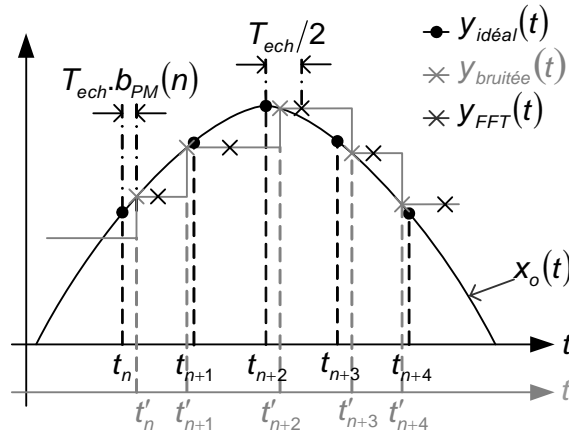


Figure 2 - 12 : Echantillonnage pour le calcul de la FFT

Le jitter de l'horloge étant bien inférieur à la demi-période d'échantillonnage, cette acquisition en vue de la FFT prend en compte les fluctuations d'amplitude des échantillons mais pas les fluctuations sur l'instant d'échantillonnage t'_n .

Nous obtenons donc la fonction notée $y_{FFT}(t)$ issue d'un second échantillonnage. Le développement suivant ne tiendra pas compte du décalage d'une demi-période :

$$\begin{aligned}
 y_{FFT}(t) &= \sum_{n=0}^{n=\infty} x_o(t_n) \cdot \delta(t-t_n) \\
 &\approx \underbrace{x_o(t) \cdot \sum_{n=0}^{n=\infty} \delta(t-t_n)}_{y_{idéale}(t)} + \underbrace{T_{ech} \frac{\partial x_o}{\partial t}(t) \cdot b_{PM}(t) \cdot \sum_{n=0}^{n=\infty} \delta(t-t_n)}_{b_{ech_PM}(t)} \\
 &\approx y_{idéale}(t) + b_{ech_PM}(t)
 \end{aligned}
 \tag{Eq. 2 - 31}$$

Comme précédemment (§II.3.2), pour calculer la puissance du bruit d'échantillonnage, P_{ech_PM} , nous considérons que $\frac{\partial x_o}{\partial t}|_t$ (signal déterministe) et que $b_{PM}(t)$ (signal aléatoire) sont indépendants. La puissance du bruit est alors égale à [MPR99] (cf. Eq. 2 - 25) :

$$\begin{aligned}
 P_{ech_PM} &= E\left(\left| T_{ech} \cdot b_{PM}(t) \cdot \frac{\partial x_o}{\partial t} \right|^2 \right) \\
 &= (T_{ech})^2 \cdot \left[\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |b_{PM}(t)|^2 dt \right] \cdot \left[\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |2 \cdot \pi \cdot f_o \cdot A \cdot \cos(2 \cdot \pi \cdot f_o \cdot t)|^2 dt \right] \\
 &= (T_{ech})^2 \cdot \sigma_{PM}^2 \cdot \frac{(2 \cdot \pi \cdot f_o \cdot A)^2}{2} \quad \text{en } V_{rms}^2
 \end{aligned}
 \tag{Eq. 2 - 32}$$

La DSP s'écrit alors :

$$S_{ech_PM}(f)_{V_{rms}^2/Hz} = (\sigma \cdot T_{ech} \cdot 2 \cdot \pi \cdot f_o \cdot A)^2 / 2 \cdot F_{ech} \quad \text{Eq. 2 - 33}$$

Nous retrouvons ici une relation communément admise pour une première évaluation de l'effet de jitter sur les circuits à temps discret [MPR99].

La DSP du bruit d'échantillonnage en dB_c/Hz , notée $S_{ech_PM}(f)_{dB_c/Hz}$, est égale à :

$$S_{ech_PM}(f)_{dB_c/Hz} = 20 \cdot \log\left(\frac{2 \cdot \pi \cdot f_o}{F_{ech}}\right) + S_{b_{PM}}(f)_{dB_v/Hz} \quad \text{Eq. 2 - 34}$$

N.B. : nous verrons dans le paragraphe suivant que cela n'est pas totalement exact.

II.3.4. Généralisation

Jusqu'à présent, afin de faciliter les calculs de puissance, nous nous sommes limités à l'évaluer l'influence d'une seule source de bruit blanc.

Nous considérons maintenant que les sources de bruits $b_A(t)$ et $b_{PM}(t)$ sont issues de l'addition d'une source de bruit blanc et d'une source de bruit de Flicker. Elles ont alors un spectre dont les propriétés sont énoncées dans l'Annexe 1 (cf. §II.1.1) et qui est représenté Figure 2 - 13. Les DSP respectives en dB_v/Hz de ces deux bruits seront toujours notées : $S_{b_A}(f)_{dB_v/Hz}$ et $S_{b_{PM}}(f)_{dB_v/Hz}$.

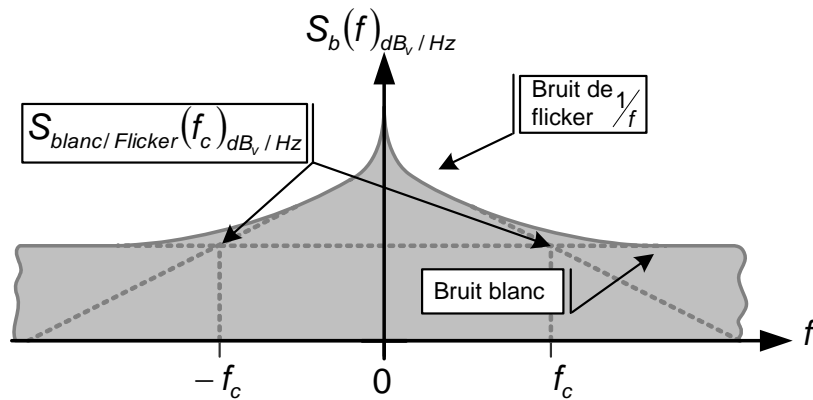


Figure 2 - 13 : DSP du bruit $b_A(t)$ (ou $b_{PM}(t)$)

Nous cherchons maintenant à obtenir la relation liant la DSP d'une sinusoïde d'amplitude bruitée $S_x(f)_{dB_c/Hz}$ et la DSP du bruit d'amplitude à l'origine de cet dégradation $S_{b_A}(f)_{dB_v/Hz}$. Il en sera de même pour la relation entre la DSP du bruit d'échantillonnage PM d'une sinusoïde idéale $S_{ech_PM}(f)_{dB_c/Hz}$ et la DSP du jitter PM $S_{b_{PM}}(f)_{dB_v/Hz}$.

Le bruit d'une sinusoïde d'amplitude bruitée et le bruit d'échantillonnage PM d'une sinusoïde idéale sont tous les deux des bruits modulés par rapport à une porteuse :

$$x(t) \approx x_o(t) + \underbrace{b_A(t).A}_{\text{bruit d'amplitude}} \cdot \sin(2.\pi.f_o.t) \quad (\text{Eq.2.23})$$

$$y_{FFT}(t) \approx y_{\text{ech idéale}}(t) + \underbrace{b_{PM}(t).A.2.\pi.f_o.F_{\text{ech}}}_{\text{bruit d'échantillonnage}} \cdot \cos(2.\pi.f_o.t) \cdot \sum_{n=0}^{n=\infty} \delta(t-t_n) \quad (\text{Eq.2.31}) \quad \text{Eq. 2 - 35}$$

Si nous référons à l'Annexe 3 (cf. §II.3.1) avec $\alpha = 1$ pour le bruit d'une sinusoïde d'amplitude bruitée nous pouvons écrire :

$$S_x(\pm f_o \pm f)_{dB_c/Hz} = \begin{cases} S_{b_A(\text{Flic})}(\pm f)_{dB_v/Hz} & -\frac{f_c}{2} < \pm f < \frac{f_c}{2} \\ S_{b_A(\text{Flic}/\text{blc})}(\pm f_c)_{dB_v/Hz} + 10.\log(4) & f = \pm \frac{f_c}{2} \\ S_{b_A(\text{blc})}(f_c)_{dB_v/Hz} + 10.\log(2) & \text{sinon} \end{cases} \quad \text{Eq. 2 - 36}$$

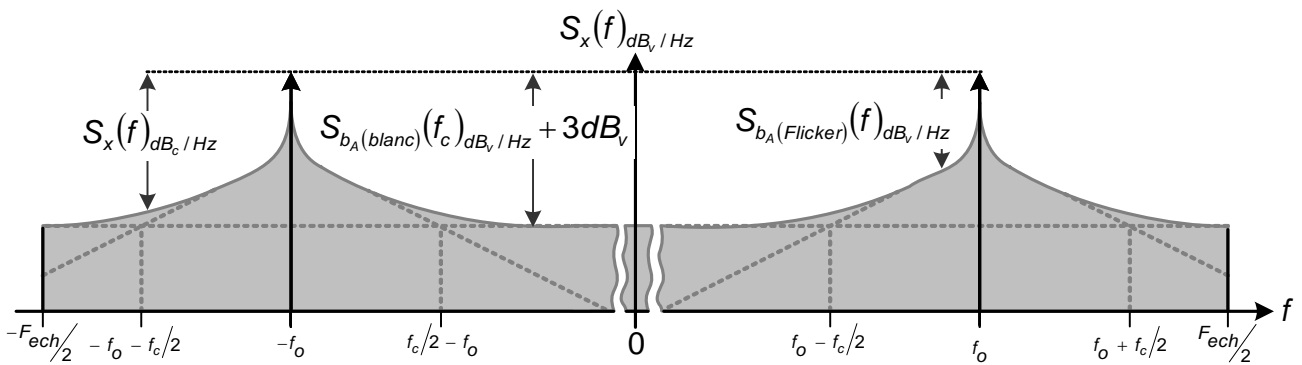


Figure 2 - 14 : DSP du bruit d'un sinusoïde d'amplitude bruitée

Et avec $\alpha = 2.\pi.f_o/F_{\text{ech}}$ pour le bruit d'échantillonnage PM nous pouvons écrire :

$$S_{\text{ech_PM}}(\pm f_o \pm f)_{dB_c/Hz} = 20.\log\left(\frac{2.\pi.f_o}{F_{\text{ech}}}\right) + \begin{cases} S_{b_{PM}(\text{Flic})}(\pm f)_{dB_v/Hz} & -\frac{f_c}{2} < \pm f < \frac{f_c}{2} \\ S_{b_{PM}(\text{Flic}/\text{blc})}(\pm f_c)_{dB_v/Hz} + 10.\log(4) & f = \pm \frac{f_c}{2} \\ S_{b_{PM}(\text{blc})}(f_c)_{dB_v/Hz} + 10.\log(2) & \text{sinon} \end{cases} \quad \text{Eq. 2 - 37}$$

N.B. : Nous constatons que les résultats des Eq. 2 - 28 et Eq. 2 - 34 sont erronés car nous avons négligé les effets de recouvrement de spectre lors de la modulation d'un bruit blanc décrit dans l'Annexe 2 (cf §II.2.3 et §II.2.4).

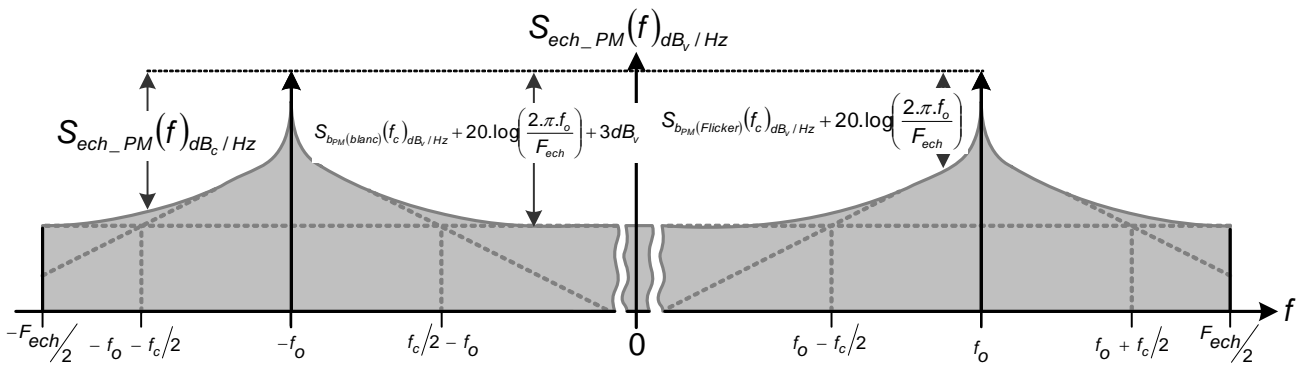


Figure 2 - 15 : DSP du bruit d'échantillonnage PM

II.4. Bruit de phase d'un oscillateur

Intéressons nous maintenant au bruit de phase qui affecte les oscillateurs autonomes (VCO, oscillateur asynchrone,...). Ce bruit de phase se traduit dans le domaine temporel par le jitter FM, introduit par K. Kundert [Kun98]. Il désigne la variation de la phase, issue de l'accumulation de variations aléatoires à chaque transition de l'oscillateur. C'est un bruit cumulatif. Le jitter FM est à opposer au jitter PM (Phase Modulation) (cf. §II.3.3) qui, lui, affecte les générateurs pilotés et désigne une variation des instants de transition indépendante des instants précédents. Jitter et bruit de phase sont deux expressions différentes du même phénomène aléatoire. Il est donc possible de passer de l'un à l'autre. C'est ce que nous allons développer dans les prochains paragraphes.

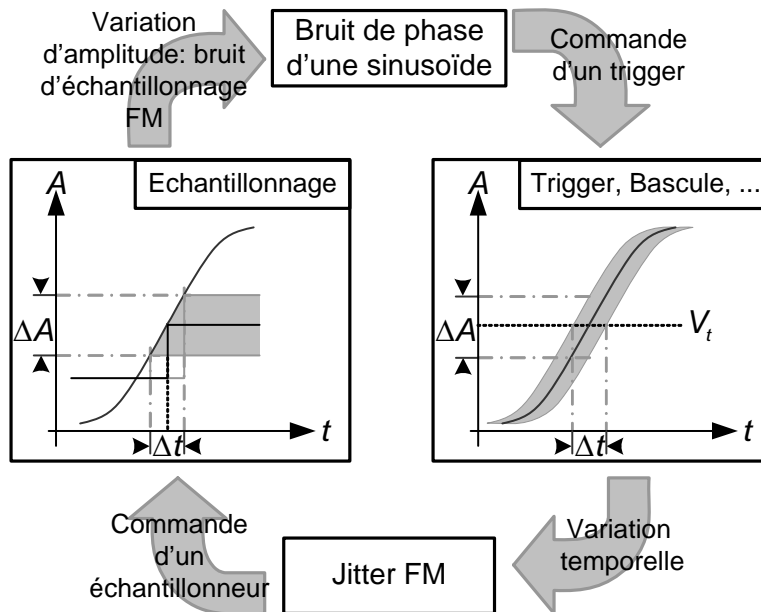


Figure 2 - 16 : Relation entre le jitter FM et le bruit de phase d'un générateur de sinusoïdes

N.B. : il existe aussi des relations semblables entre le jitter PM et le bruit d'une sinusoïde d'amplitude bruitée.

II.4.1. Origine du bruit de phase

Considérons un VCO (Voltage Controlled Oscillator) dont la fréquence de sortie est proportionnelle à la tension d'entrée :

$$F_{out} = K.V_{in} \quad \text{Eq. 2 - 38}$$

Si, comme illustré ci-dessous, le générateur de tension est bruité (bruit thermique, de grenaille, de Flicker), alors la fréquence de sortie sera soumise à une variation $\Delta f(t)$. Il en va de même pour les bruits internes au VCO qui peuvent être ramenés en entrée par une source de bruit équivalente [Nei01] :

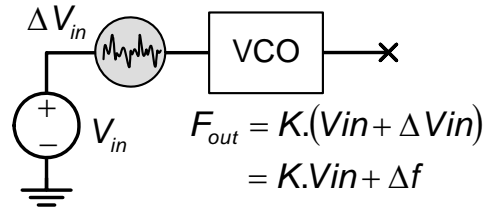


Figure 2 - 17 : VCO avec entrée bruitée

La variation de phase de cet oscillateur est liée à variation de la fréquence par :

$$\Delta\phi_{osc}(t) = 2.\pi. \int \Delta f(u).du \quad \text{Eq. 2 - 39}$$

Nous notons $S_{\Delta\phi_{osc}}(f)_{V_{rms}^2/Hz}$ la DSP de la phase $\Delta\phi_{osc}(t)$ en V_{rms}^2/Hz , et $S_{\Delta f}(f)_{V_{rms}^2/Hz}$ la DSP de la variation de fréquence $\Delta f(t)$ en V_{rms}^2/Hz . [BaA66] et [CuS66] ont démontré la relation suivante :

$$\underbrace{\Delta\phi_{osc}(t) = 2.\pi. \int \Delta f(u).du}_{\text{domaine temporel}} \Leftrightarrow \underbrace{S_{\Delta\phi_{osc}}(f)_{V_{rms}^2/Hz}}_{\text{domaine fréquentiel}} = \frac{S_{\Delta f}(f)_{V_{rms}^2/Hz}}{|j.2.\pi.f|^2} \quad \text{Eq. 2 - 40}$$

Autrement dit, si ΔV_{in} est un bruit blanc alors $S_{\Delta\phi_{osc}}(f)_{V_{rms}^2/Hz}$ sera proportionnel à $1/f^2$ et si ΔV_{in} est un bruit de Flicker ($1/f$) alors $S_{\Delta\phi_{osc}}(f)_{V_{rms}^2/Hz}$ sera proportionnel à $1/f^3$

En considérant maintenant les DSP en dB_v/Hz de $\Delta\phi_{osc}(t)$ et $\Delta f(t)$ notées respectivement $S_{\Delta\phi_{osc}}(f)_{dB_v/Hz}$ et $S_{\Delta f}(f)_{dB_v/Hz}$, nous pouvons écrire :

$$S_{\Delta\phi_{osc}}(f)_{dB_v/Hz} = S_{\Delta f}(f)_{dB_v/Hz} - 20.\log(2.\pi.f) \quad \text{Eq. 2 - 41}$$

Cette transformation entre $S_{\Delta\phi_{osc}}(f)_{dB_v/Hz}$ et $S_{\Delta f}(f)_{dB_v/Hz}$ est illustrée ci-dessous :

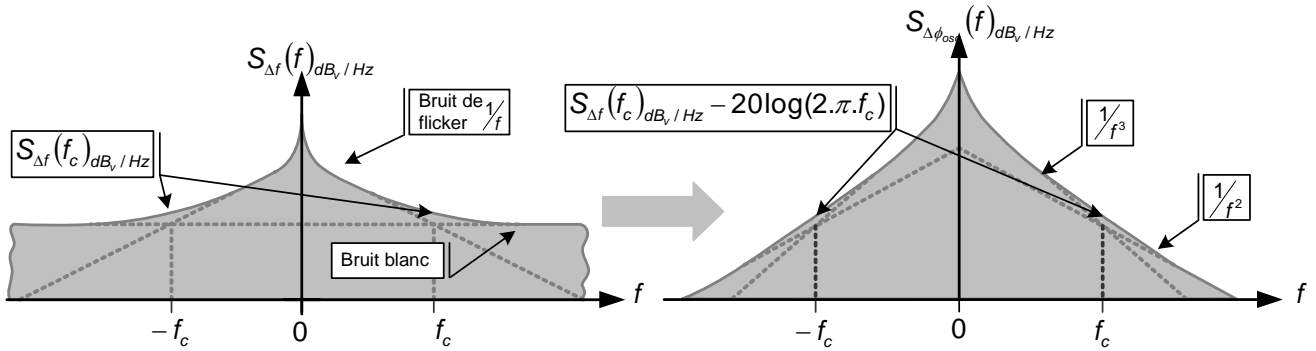


Figure 2 - 18 : Transformation d'un bruit blanc et de Flicker en $1/f^2$ et $1/f^3$

N.B. : Cette propriété nous sera très utile pour la modélisation de ces bruits.

II.4.2. Spectre du bruit de phase d'une sinusoïde

Nous nous intéressons aux variations d'amplitude d'une sinusoïde dues aux variations de sa phase. Considérons pour cela une porteuse soumise à une variation de sa phase : $x(t) = A \cdot \sin(2\pi \cdot f_o \cdot t + \Delta\phi_x(t))$. L'excès de phase $\phi_x(t)$ a une DSP en dB_v/Hz notée $S_{\phi_x}(f)_{dB_v/Hz}$ qui a une variation en $1/f^2$ et $1/f^3$. Soit $x_o(t) = A \cdot \sin(2\pi \cdot f_o \cdot t)$ la porteuse idéale.

Comme nous considérons que les variations de la phase sont petites devant 1, nous pouvons effectuer l'approximation suivante :

$$\begin{aligned}
 x(t) &= A \cdot \sin(2\pi \cdot f_o \cdot t + \Delta\phi_x(t)) \\
 &= A \cdot \sin(2\pi \cdot f_o \cdot t) \cdot \cos(\Delta\phi_x(t)) + A \cdot \cos(2\pi \cdot f_o \cdot t) \cdot \sin(\Delta\phi_x(t)) \\
 &\approx A \cdot \sin(2\pi \cdot f_o \cdot t) + \Delta\phi_x(t) \cdot A \cdot \cos(2\pi \cdot f_o \cdot t) \quad \text{car } \Delta\phi_s(t) \ll 1 \\
 &\approx x_o + \Delta A \left(\Delta\phi_x, \frac{\partial x}{\partial t} \right)
 \end{aligned}
 \tag{Eq. 2 - 42}$$

La sinusoïde avec la phase bruitée peut alors s'apparenter à la somme d'une sinusoïde non bruitée et d'un bruit d'amplitude ΔA dépendant de la variation de phase et de la dérivée de la sinusoïde. Conformément à l'étude menée dans l'Annexe 3 (cf. §II.3.2) la DSP du bruit de phase de la sinusoïde en dB_c/Hz notée $S_x(f)_{dB_c/Hz}$ est alors égale à :

$$\begin{aligned}
 S_x(f \pm f_o)_{dB_c/Hz} &\approx 10 \cdot \log \left(\frac{S_{\Delta A}(f)_{V_{rms}^2/Hz}}{P_{x_o}(f_o)_{V_{rms}^2}} \right) \\
 &\approx S_{\Delta\phi_x}(f)_{dB_v/Hz}
 \end{aligned}
 \tag{Eq. 2 - 43}$$

La figure suivante illustre la répartition du bruit de phase autour de la porteuse :

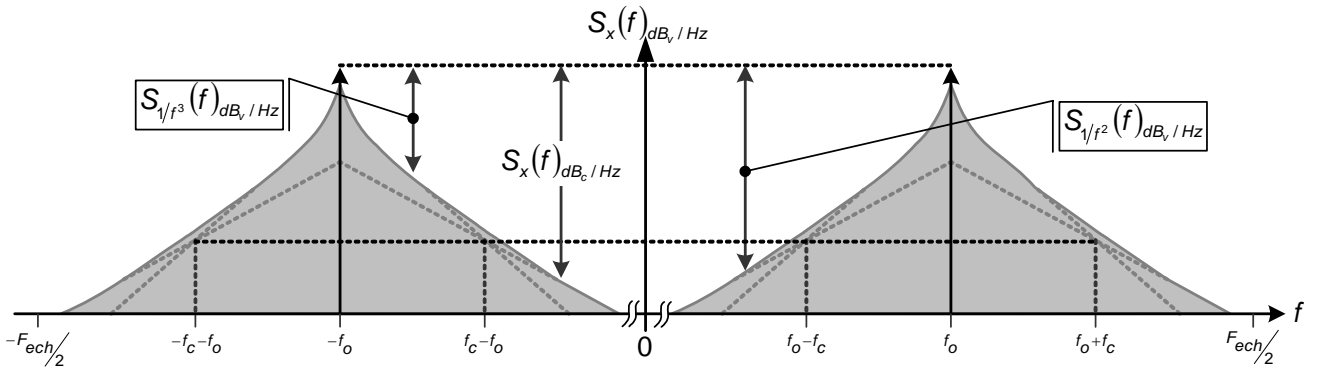


Figure 2 - 19 : Bruit de phase d'une porteuse

N.B. : Si nous considérons maintenant, comme illustré par la Figure 2 - 16, que cette sinusoïde sert de générateur d'horloge (sur les fronts montants par exemple), alors nous pouvons établir une relation entre le bruit de phase de la sinusoïde et le jitter FM. Nous noterons V_t le seuil de comparaison.

$$V_t = A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t_n + \phi_x) \Rightarrow t_n = \frac{1}{2 \cdot \pi \cdot f_o} \cdot \left(\arcsin\left(\frac{V_t}{A}\right) - \phi_x \right) \quad \text{Eq. 2 - 44}$$

$$\Rightarrow t_n = \underbrace{\left(\frac{1}{2 \cdot \pi \cdot f_o} \right) \cdot \arcsin(V_t / A)}_{\text{instant idéal}} - \underbrace{\left(\frac{\phi_x}{2 \cdot \pi \cdot f_o} \right)}_{\Delta t_{\text{jitter FM}}}$$

Si $V_t = 0$ alors :

$$t_n = n \cdot T_o - \Delta t_{\text{jitter FM}} \quad \text{avec} \quad \Delta t_{\text{jitter FM}} = \frac{\phi_x}{2 \cdot \pi \cdot f_o} \quad \text{Eq. 2 - 45}$$

II.4.3. Influence du jitter FM sur l'échantillonnage d'une porteuse

Considérons maintenant une porteuse idéale $x_o(t) = A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t)$ échantillonnée par une horloge qui est sujette au jitter FM. Comme nous nous intéressons aux variations temporelles nous changeons de notation pour la phase de l'horloge que nous désignerons par $\Delta \phi_{FM}(t)$. Sa DSP dB_V/Hz est notée $S_{\Delta \phi_{FM}}(f)_{dB_V/Hz}$. Le nouvel instant d'échantillonnage est alors égal à :

$$2 \cdot \pi \cdot F_{ech} \cdot t_n'' + \Delta \phi_{FM}(t_n'') = n \cdot 2 \cdot \pi \quad \text{soit} \quad t_n'' = T_{ech} \cdot \left(n - \frac{\Delta \phi_{FM}(t_n'')}{2 \cdot \pi} \right) \quad \text{Eq. 2 - 46}$$

Ainsi l'échantillonnage de la porteuse s'exprimera (§ II.3.3) :

$$y_{\text{bruitée}}(t) = \sum_{n=0}^{n=\infty} A \cdot \sin(2 \cdot \pi \cdot f_o \cdot n \cdot T_{ech} - T_{ech} \cdot f_o \cdot \Delta \phi_{FM}(t_n'')) \cdot \delta(t - t_n'')$$

$$= \sum_{n=0}^{n=\infty} x_o(t_n'') \cdot \delta(t - t_n'') \quad \text{Eq. 2 - 47}$$

Nous utilisons la fonction $y_{FFT}(t)$ définie précédemment (Figure 2 - 12 et Eq. 2 - 33) en considérant que la dérive de l'instant de l'échantillonnage n'excède pas $T_{ech}/2$:

$$y_{FFT}(t) = \sum_{n=0}^{n=\infty} x(t_n) \cdot \delta(t - t_n) \quad \text{Eq. 2 - 48}$$

On constate que cet échantillonnage bruité équivaut à l'introduction d'un excès de phase $\Delta\phi'_x(t)$ sur la porteuse :

$$x'(t) = A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t + \Delta\phi'_x(t)) \quad \text{avec} \quad \Delta\phi'_x(t) = -T_{ech} \cdot f_o \cdot \Delta\phi_{FM}(t) \quad \text{Eq. 2 - 49}$$

Nous pouvons donc caractériser comme précédemment le bruit de phase de cette nouvelle porteuse par le rapport entre la puissance de la porteuse idéale et le bruit autour de la porteuse. Soit d'un point de vue des DSP :

$$S_{x'}(|f_o \pm f_m|)_{dB_c/Hz} \approx S_{\Delta\phi'_x}(f_m)_{dB_v/Hz} \quad \text{Eq. 2 - 50}$$

Avec la relation suivante :

$$S_{\Delta\phi'_x}(f_m)_{dB_v/Hz} = 20 \cdot \log\left(\frac{f_o}{F_{ech}}\right) + S_{\Delta\phi_{FM}}(f_m)_{dB_v/Hz} \quad \text{Eq. 2 - 51}$$

Nous introduisons alors la notion de bruit d'échantillonnage dû au jitter FM, qui se traduit en fait par le bruit de phase de la porteuse $x'(t)$. Nous notons sa DSP en dB_c/Hz : $S_{ech_FM}(f)_{dB_c/Hz}$:

$$S_{ech_FM}(f \pm f_o)_{dB_c/Hz} \approx 20 \cdot \log\left(\frac{f_o}{F_{ech}}\right) + S_{\Delta\phi_{FM}}(f)_{dB_v/Hz} \quad \text{Eq. 2 - 52}$$

Nous disposons maintenant de l'ensemble des relations théoriques qui lient l'expression du bruit dans le domaine temporel ($b_A(t)$, $b_{PM}(t)$, $\Delta\phi_x(t)$ et $\Delta\phi_{FM}(t)$) et le domaine fréquentiel ($S_{b_A}(f)_{dB_v/Hz}$, $S_{b_{PM}}(f)_{dB_v/Hz}$, $S_{\Delta\phi_x}(f)_{dB_v/Hz}$ et $S_{\Delta\phi_{FM}}(f)_{dB_v/Hz}$) et ainsi que les relations avec : $S_x(f)_{dB_c/Hz}$, $S_{ech_PM}(f)_{dB_c/Hz}$ et $S_{ech_FM}(f)_{dB_c/Hz}$. Nous pouvons maintenant nous consacrer à la modélisation de ces bruits sous forme de fonctions. Nous utiliserons les expressions des différentes DSP pour valider nos modèles.

III. MODELISATION DES BRUITS

III.1. Le générateur de gaussienne

III.1.1. Principe

Toutes nos sources de bruit sont basées sur un générateur de nombres aléatoires de densité de probabilité gaussienne. Or aucune fonction en VHDL-AMS ne fournit ce type de distribution. Nous avons donc dû en créer une à partir des fonctions de nombres aléatoires déjà existantes. Nous avons pour cela utilisé la fonction UNIFORM qui fournit des nombres pseudo aléatoires avec une répartition uniforme entre $]0;1]$. La transformée la plus connue pour passer d'une répartition uniforme à une répartition gaussienne est celle de Box Muller [BoM58] : soient (u_1, u_2) des fonctions aléatoires avec une répartition uniforme entre $]0;1]$, alors v est à répartition gaussienne de moyenne m et d'écart type égal σ si :

$$v = m + \sigma \cdot \sqrt{-2 \cdot \text{Log}(u_1)} \cdot \sin(2 \cdot \pi \cdot u_2) \quad \text{Eq. 2 - 53}$$

III.1.2. Modèle

Nous avons donc utilisé le principe précédemment énoncé pour modéliser notre source gaussienne de nombres aléatoires. Le code est présenté ci-dessous :

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

L'utilisateur entre la valeur de l'écart type et un vecteur à 5 composantes (contenant les états précédents du générateur de nombres aléatoires).

Seul le vecteur seed(1 TO 4) est utilisé, il correspond aux états précédents des générateurs de nombres pseudo aléatoires. Ces états précédents peuvent être stockés dans un fichier cf. [WaH04]

Nous utilisons la fonction UNIFORM de la bibliothèque math_real Eq. 2 - 53 avec $m=0$. La fonction retourne un nombre gaussien rand(0). Et renvoie les états des générateurs de nombres pseudo aléatoires : rand(1 TO 4)

Code 2 - 1 : Modèle en VHDL-AMS de la fonction génératrice de nombres aléatoires gaussiens

N.B. : La fonction génératrice de nombres aléatoires est de moyenne nulle afin de ne pas générer de composante DC pour les bruits.

III.1.3. Performances

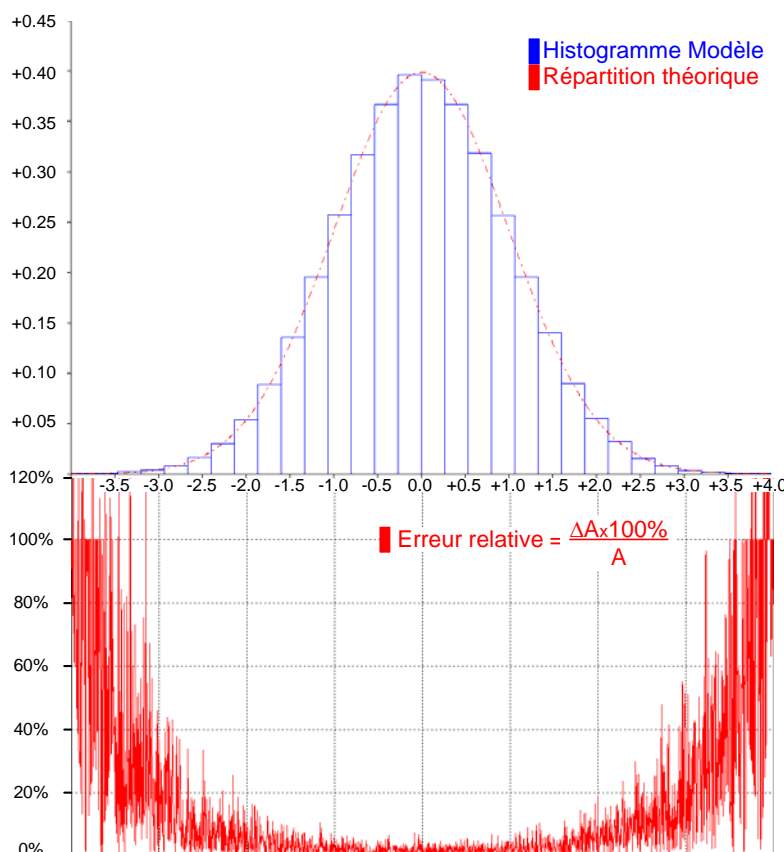
Afin de qualifier notre générateur de nombres aléatoires gaussien, nous avons eu recours à deux méthodes différentes d'estimation de la moyenne et de l'écart-type. Pour cela, nous avons simulé notre modèle (avec les paramètres `std_deviation=1`) à l'aide du logiciel ADVanceMS [ADV00] et extrait 10^5 échantillons.

Nous avons alors calculé la moyenne et l'écart-type en appliquant les fonctions suivantes [DeQ96] :

$$m = \frac{1}{N} \sum_{n=0}^{n=N-1} gauss(n) \quad \left| \quad \sigma = \sqrt{\frac{1}{N} \sum_{n=0}^{n=N-1} (m - gauss(n))^2} \right. \quad \text{Eq. 2 - 54}$$

Nous trouvons alors $m \approx 0.0014$ et $\sigma \approx 1.00014$.

Une seconde méthode utilise le même fichier de points et le traite en utilisant le logiciel SCILAB [SCI]. Nous obtenons alors la répartition des échantillons sous forme d'histogramme ainsi que l'erreur relative entre l'histogramme et la répartition théorique d'un gaussienne :



Nombre d'échantillons: 10^5
obtenus avec le générateur gaussien

On trouve :

- 68.007% entre $[m-\sigma, m+\sigma]$
- 95.478% entre $[m-2\sigma, m+2\sigma]$
- 99.685% entre $[m-3\sigma, m+3\sigma]$

Figure 2 - 20 : Histogramme et erreur relative à partir de 10^5 échantillons

Les propriétés statistiques mesurées du générateur ainsi conçu sont donc conformes à celles d'un générateur gaussien.

III.2. Le bruit blanc de tension

III.2.1. Principe

Ce modèle est une extension du précédent modèle : nous appliquons la déviation gaussienne à une quantité de type tension. Puisque les appels récurrents à la fonction $gauss(seed, \sigma)$ sont indépendants les uns des autres, le bruit généré est bien un bruit blanc. La fréquence de répétition est F_{ech} (Eq. 2 - 21, Eq. 2 - 22) :

$$\sigma_{blanc} = \sqrt{F_{ech} \cdot 50.1e^{-3} \cdot 10^{\frac{S_{blanc}(f)_{dBm/Hz}}{10}}}$$

$$\text{ou } \sigma_{blanc} = \sqrt{F_{ech} \cdot 10^{\frac{S_{blanc}(f)_{dBv/Hz}}{10}}}$$

Eq. 2 - 55

N.B. : L'écart-type σ_{blanc} s'identifie, dans ce cas à la valeur V_{rms} du bruit blanc de tension.

III.2.2. Modèle

Tous les bruits sont modélisés sous forme de fonctions en VHDL-AMS. Pour simuler toutes les fonctions (§III.2.2, III.3.2 et III.4.2) nous devons créer un modèle qui réalisera les appels successifs à la fréquence F_{ech} de ces fonctions. Nous donnons le code de ce modèle ci-dessous :

<pre> ----- ENTITY test_mod IS GENERIC (Fech : REAL := 20.0e+6; PSD_dBv_Hz : REAL := -80.0); PORT (SIGNAL : Nbruit : OUT REAL := 0.0); END ENTITY test_mod; ----- ARCHITECTURE behaviour OF test_mod IS BEGIN Output_generation : PROCESS VARIABLE rand_white : REAL_VECTOR (0 TO 4) := (0.0,27.0,56.0,456.0,64.0); BEGIN rand_white := white_noise (Fech, PSD_dBv_Hz, rand_white); Nbruit <= rand_white(0) ; WAIT FOR 1.0/Fech ; END PROCESS Output_generation; END ARCHITECTURE behaviour; ----- </pre>	<p><i>L'utilisateur entre la fréquence d'échantillonnage et la valeur de DSP_dBv_Hz.</i></p> <p><i>La sortie du modèle s'effectue par un port de type SIGNAL</i></p> <p><i>Début du PROCESS</i></p> <p><i>Conditions initiales pour les générateurs de nombres aléatoires</i></p> <p><i>Appel de la fonction de bruit (cf. Code 2 - 3)</i></p> <p><i>Répétition du PROCESS à la fréquence Fech.</i></p>
---	--

Code 2 - 2 : Modèle en VHDL-AMS d'appel des fonctions de bruit

La fonction bruit blanc ne contient que la fonction de génération de nombres gaussiens $gauss(seed, \sigma)$ et une relation de passage de l'écart-type à la DSP de bruit :

```

-----
FUNCTION white_noise_dBm (Fech, PSD_dBm_Hz : REAL; seed:
                        REAL_VECTOR) RETURN REAL_VECTOR IS
CONSTANT std_dev : REAL := SQRT(Fech*50.0*1.0e-3*10**(PSD_dBm_Hz/10.0));
VARIABLE rand    : REAL_VECTOR (0 TO 4);
BEGIN
    rand := gauss(std_dev,seed);
RETURN (rand);
END FUNCTION;
-----
FUNCTION white_noise (Fech, PSD_dBv_Hz : REAL; seed: REAL_VECTOR)
                    RETURN REAL_VECTOR IS
CONSTANT std_dev    : REAL := SQRT(Fech*10**(PSD_dBv_Hz/10.0));
VARIABLE rand       : REAL_VECTOR (0 TO 4);
BEGIN
    rand := gauss(std_dev,seed);
RETURN (rand);
END FUNCTION;
-----

```

L'utilisateur entre la fréquence d'échantillonnage la valeur de DSP_dBm_Hz ou DSP_dBv_Hz et un vecteur à 5 composantes.

Eq. 2 - 55

La fonction retourne rand(0) de DSP égale à : DSP_dBm_Hz ou DSP_dBv_Hz

Code 2 - 3 : Modèle en VHDL-AMS de la fonction génératrice de bruit blanc

III.2.3. Performances

La vérification des caractéristiques du modèle du générateur de bruits a nécessité diverses simulations effectuées grâce au logiciel ADVanceMS [ADV00]. Nous simulons le modèle *white_noise* avec pour paramètres : $F_{ech} = 20MHz$ et $S_{blanc}(f)_{dB_v/Hz} = -80 dB_v / Hz$.

Le traitement fréquentiel est réalisé grâce au logiciel SCILAB associé au programme décrit (§II.1.3). Nous disposons d'un nombre total d'échantillons $N_{ech} = 2^{20}$, le périodogramme est réalisé sur des sous fenêtres de 2^{14} échantillons avec une fenêtre rectangulaire soit une bande de bruit équivalente de $BBE = 1220.7Hz$. Nous obtenons alors la

Nous mesurons une DSP de bruit blanc de :

$$\begin{aligned}
 S_{blanc}(f)_{dB_v/Hz} &= -49dB_v / \Delta f - 10.Log(1220.7) \\
 &\approx -80 dB_v / Hz
 \end{aligned}$$

Eq. 2 - 56

La simulation corrobore l'étude théorique.

N.B. : On peut également exprimer le dernier résultat en dB_m / Hz :

$$S_{blanc}(f)_{dB_m/Hz} \approx -67 dB_m / Hz.$$

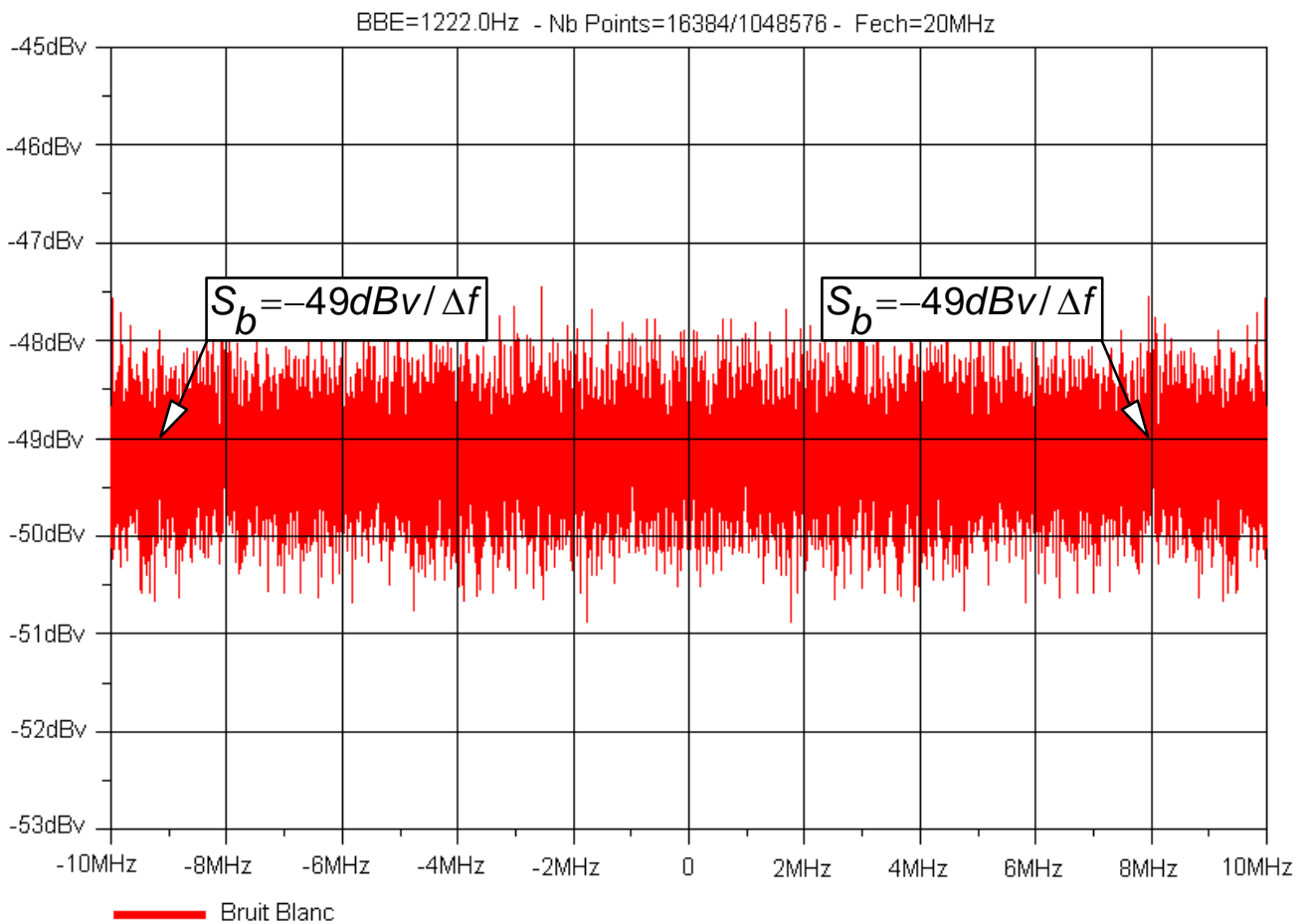


Figure 2 - 21 : Spectre du bruit de tension blanc

III.3. Le bruit de tension en $1/f$

III.3.1. Principe

Pour modéliser le bruit de Flicker ($1/f$), nous utilisons le générateur de bruit blanc précédemment synthétisé auquel nous appliquons un filtre en z qui nous fournit une pente de -10 dB/décade sur une bande de fréquence de 100Hz à 1MHz. Nous utilisons pour la synthèse du filtre en z la méthode des filtres d'ordre non entier [Ous83], [Ous95] que nous avons détaillée dans l'Annexe 4 (cf. §II.4). Cette méthode consiste en la mise en cascade de filtre d'ordre entier afin d'obtenir un filtre global d'ordre non-entier. Les paramètres généraux ont été définis en Annexe, nous allons maintenant procéder à la définition des paramètres propres au modèle.

Les travaux développés dans cette partie ont fait l'objet de 2 articles [LMS04] et [MLD03].

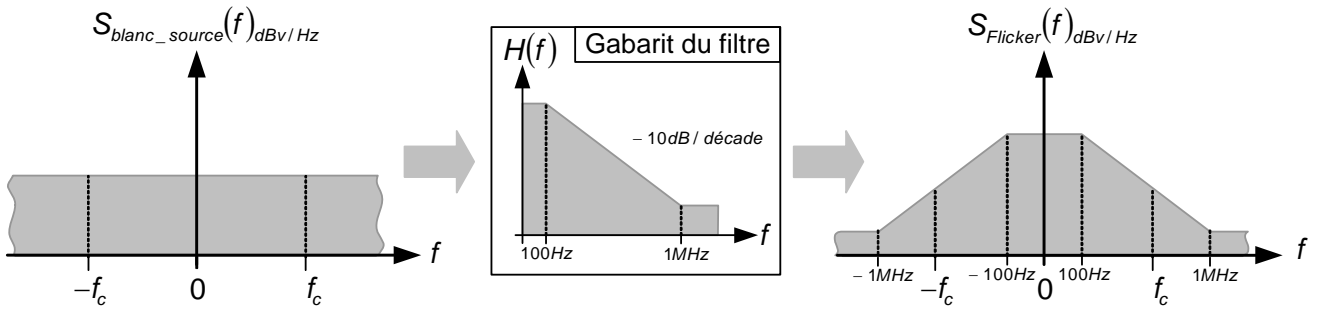


Figure 2 - 22 : Principe de synthèse du bruit de tension en 1/f

Après de nombreuses simulations, nous avons constaté que l'écart entre le temps de simulation du filtre le plus précis (avec $\beta^2 = 2$) et le temps de simulation du filtre le moins précis (avec $\beta^2 = 8$) n'était pas conséquent. Nous avons donc privilégié la précision du modèle et choisi les paramètres suivants pour le filtre : $\beta^2 = 2$, $\alpha = 467.77m$, $\gamma = 4.57$, $n = 7$ et $f_m^i = \gamma^i \cdot 100Hz$.

Comme défini précédemment dans l'Annexe 1 (cf. §II.1.1), le bruit de Flicker est caractérisé par le couple $(f_c; S_b(f_c)_{dB_v/Hz})$ tel que :

$$S_{Flicker}(f_c)_{dB_v/Hz} = S_{blanc}(f_c)_{dB_v/Hz} \quad \text{Eq. 2 - 57}$$

Or le bruit de Flicker est issu du filtrage d'un bruit blanc source (cf. Figure 2 - 22), d'où :

$$S_{Flicker}(f_c)_{dB_v/Hz} = S_{blanc_source}(f_c)_{dB_v/Hz} + 10 \cdot \log(|H(f_c)|^2) \quad \text{Eq. 2 - 58}$$

Donc, pour avoir l'égalité entre Eq. 2 - 57 et Eq. 2 - 58, il faut que $|H(f_c)|^2 = 1$. Pour cela les H_o^i doivent vérifier :

$$|H_o^i|^2 = \frac{1 + (2 \cdot \pi \cdot f_c \cdot \tau_i)^2}{1 + (2 \cdot \pi \cdot f_c \cdot \alpha \cdot \tau_i)^2} \quad \forall i \in [0 \dots n-1] \Rightarrow |H(f_c)|^2 = \left| \prod_{j=0}^{j=n-1} H_o^j \cdot \frac{1 + j \cdot 2 \cdot \pi \cdot f_c \cdot \alpha \cdot \tau_j}{1 + j \cdot 2 \cdot \pi \cdot f_c \cdot \tau_j} \right|^2 = 1 \quad \text{Eq. 2 - 59}$$

avec $\tau_i = (2 \cdot \pi \cdot \sqrt{\alpha} \cdot f_m^i)^{-1}$

Comme les signaux étudiés sont discrets, nous utilisons une transformation bilinéaire pour convertir notre fonction de transfert de LAPLACE en z, utilisable dans le domaine temporel.

$$H(p) = \prod_{i=0}^{i=N-1} H_o^i \cdot \frac{1 + \alpha \tau_i p}{1 + \tau_i p} \xrightarrow{p = \frac{2}{T_{ech}} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}} H(z) = \prod_{i=0}^{i=N-1} H_o^i \cdot \frac{(T_{ech} + 2 \cdot \alpha \cdot \tau_i) + z^{-1}(T_{ech} - 2 \cdot \alpha \cdot \tau_i)}{(T_{ech} + 2 \cdot \tau_i) + z^{-1}(T_{ech} - 2 \cdot \tau_i)} \quad \text{Eq. 2 - 60}$$

III.3.2. Modèle

Le modèle de bruit de jitter, se décompose en deux fonctions décrites ci-dessous :

```

FUNCTION param_1_f_filter (Beta_2 : INTEGER) RETURN REAL_VECTOR IS
  VARIABLE param      : REAL_VECTOR (0 TO 2);
BEGIN
CASE Beta_2 IS
  WHEN 2      => param(0) := 0.4677601; param(2) := 7.0;
  WHEN 3      => param(0) := 0.4449912; param(2) := 7.0;
  WHEN 4      => param(0) := 0.4221759; param(2) := 7.0;
  WHEN 5      => param(0) := 0.4002763; param(2) := 6.0;
  WHEN 6      => param(0) := 0.3795035; param(2) := 6.0;
  WHEN 7      => param(0) := 0.3600550; param(2) := 6.0;
  WHEN 8      => param(0) := 0.3421292; param(2) := 5.0;
  WHEN 9      => param(0) := 0.3258324; param(2) := 5.0;
  WHEN 10     => param(0) := 0.3111467; param(2) := 5.0;
  WHEN OTHERS => param(0) := SQRT(0.1); param(2) := 5.0;
END CASE; param(1) := 1.0/param(0)**2;
RETURN (param);
END FUNCTION;

```

Cette fonction retourne les paramètres : α , γ et le nombre de filtre N , définis dans l'Annexe 4 en fonction de la valeur de β

$\alpha = \text{param}(0)$
 $\gamma = \text{param}(1)$
 $N = \text{param}(2)$

Code 2 - 4 : Modèle en VHDL-AMS de la fonction définissant les paramètres du filtre en 1/f

```

-----
FUNCTION flicker_noise (Fech, PSD_dBv_Hz_flicker, fc :REAL; Beta_2
:INTEGER := 2; int: REAL_VECTOR (0 TO 12)) RETURN REAL_VECTOR IS

CONSTANT Fo      : REAL := 100.0;
CONSTANT Tsmp    : REAL := 1.0/Fech;
CONSTANT param   : REAL_VECTOR (0 TO 2) := param_1_f_filter (Beta_2);
CONSTANT Alpha   : REAL := param(0);
CONSTANT Gama    : REAL := param(1);
CONSTANT N       : INTEGER := INTEGER(param(2));
VARIABLE taux    : REAL := 0.0;
VARIABLE H       : REAL := 1.0;
VARIABLE outt    : REAL_VECTOR (0 TO 12) := int;
VARIABLE i       : INTEGER := 0;
-----
BEGIN
-- randoming and filtering --
outt(8 TO 12) := white_noise (Fech,(PSD_dBv_Hz_flicker),int(8 TO 12));
outt(1) := outt(8);
outt(8) := int(8);
FOR i IN 2 TO 8 LOOP
IF i <= N+1 THEN
taux := 1.0/(MATH_2_PI*SQRT(Alpha)*Fo*Gama**(i-2));
H:=SQRT((1.0+(alpha*taux*MATH_2_PI*fc)**2)/(1.0+(taux*MATH_2_PI*fc)**2));
outt(i) := (1.0/(H2*(Tsmp+2.0*taux)))*((Tsmp+2.0*taux*alpha)*outt(i-1)
+(Tsmp-2.0*taux*alpha)*int(i-1)-(Tsmp-2.0*taux)*H*outt(i));
ELSE outt(i) := outt(i-1); END IF;
END LOOP;
-- output generation --
outt(0) := outt(8);
RETURN (outt);
END FUNCTION;
-----

```

L'utilisateur entre la fréquence d'échantillonnage, $S_{Flicker}(f_c)_{dB_v/Hz}$,

f_c , β et un vecteur à 13 composantes.

Définition des paramètres du filtre par l'appel de la fonction `param_1_f_filter` (cf. Code 2 - 4)

`int(9 TO 12)` = états des générateurs de nombres pseudo aléatoires
`int(1 TO 8)` = états précédents des filtres cascades

Nous utilisons la fonction `white_noise` définie (§III.2.2)

Calcul de τ_i (cf. Eq. 2 - 59).

On applique la transformée en z (cf. Eq. 2 - 60).

La fonction retourne `outt(0)` dont la DSP égale à $S_{Flicker}(f_c)$ en f_c et 12 autres composantes `outt(1 TO 12)` qui sont les états internes de la fonction qui seront utiles au calcul de la prochaine valeur

Code 2 - 5 : Modèle en VHDL-AMS de la fonction génératrice de bruit de Flicker

III.3.3. Performances

Pour la mesure des performances de notre modèle, nous avons toujours recours aux fonctions développées par M. ROBBE sous SCILAB (§II.1.3). La fréquence d'échantillonnage est égale à $F_{ech} = 20MHz$. Nous disposons de $N_{ech} = 2^{20}$ échantillons subdivisés, pour le périodogramme, en sous-fenêtres de 2^{14} échantillons. Nous utilisons une fenêtre rectangulaire ce qui induit une bande de bruit équivalente de $BBE = 1220.7Hz$.

Pour la simulation, nous réalisons la source de bruit décrite dans l'Annexe 1 (cf. §II.1.1) composée de deux sources de bruit, l'une blanche et l'autre de Flicker non corrélées l'une à l'autre. Les paramètres sont les suivants : $f_c = 100kHz$ et $S_{blanc}(f_c)_{dB_v/Hz} = S_{Flicker}(f_c)_{dB_v/Hz} = -80dB_v/Hz$.

Nous obtenons alors la figure suivante :

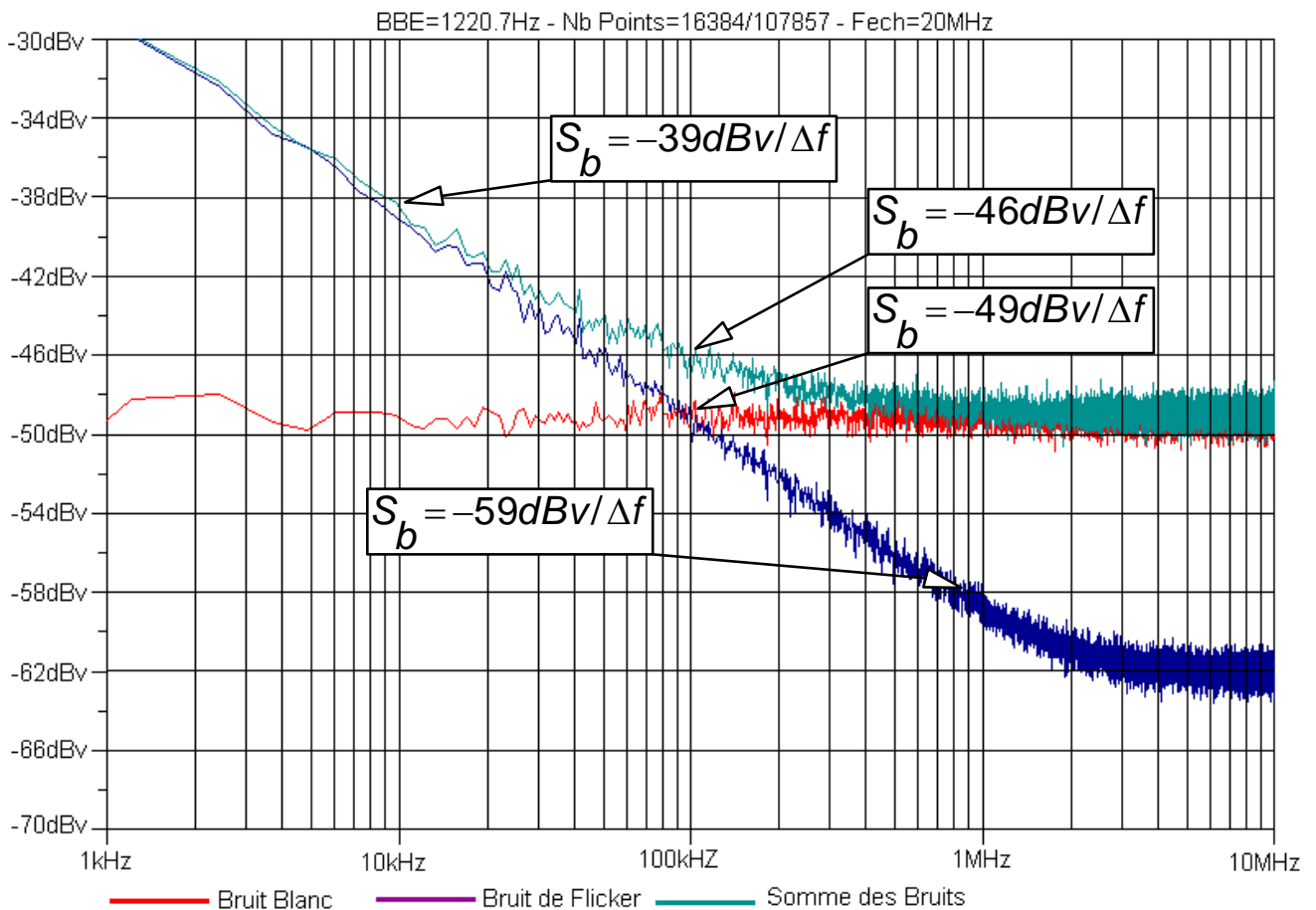


Figure 2 - 23 : Spectre du bruit de Flicker et bruit blanc

N.B. : Afin de mettre en évidence la pente de $-10 dB/décade$, nous utilisons l'échelle log pour l'axe des fréquence,, ce qui rend impossible la représentation du spectre sur l'intervalle de

fréquences $[-F_{ech}/2, F_{ech}/2]$. Nous représenterons donc le spectre sur l'intervalle de fréquences $[1kHz, F_{ech}/2]$. La courbe complète en échelle de fréquence linéaire est disponible en Annexe 5 (cf. §II.5.1)

Nous mesurons sur ce spectre les caractéristiques suivantes :

$$\begin{aligned}
 S_{bruit}(f)_{dB_v/Hz} &= S_{bruit\ mesuré}(f)_{dB_v/\Delta f} - 10 \cdot \log(1220.7) \\
 S_{blanc}(100kHz)_{dB_v/Hz} &= S_{Flicker}(100kHz)_{dB_v/Hz} \approx -80\ dB_v/Hz \\
 S_{Flicker}(10kHz)_{dB_v/Hz} &\approx -70\ dB_v/Hz \\
 S_{Flicker}(1MHz)_{dB_v/Hz} &\approx -90\ dB_v/Hz \\
 S_{somme}(100kHz)_{dB_v/Hz} &\approx -77\ dB_v/Hz
 \end{aligned}
 \tag{Eq. 2 - 61}$$

La pente du bruit de Flicker est alors de :

$$S_{Flicker}(10kHz)_{dB_v/Hz} - S_{Flicker}(1MHz)_{dB_v/Hz} = -20\ dB/2.décades
 \tag{Eq. 2 - 62}$$

La DSP de la somme des bruits de Flicker et blanc, notée $S_{b.somme}(f)$ suit la variation suivante :

$$\begin{aligned}
 S_{somme}(f) &\approx S_{Flicker}(f) \quad \text{pour } f \ll f_c \\
 S_{somme}(f) &\approx S_{blanc}(f) \quad \text{pour } f \gg f_c \\
 S_{somme}(f) &\approx \begin{cases} S_{Flicker}(f_c) \\ S_{blanc}(f_c) \end{cases} + 10 \cdot \log(2)
 \end{aligned}
 \tag{Eq. 2 - 63}$$

Les simulations sont donc conformes à nos attentes, ce qui valide notre modèle.

N.B. : Nous constatons effectivement que, au-delà de 1MHz, nous sortons du domaine de validité du filtre. La fréquence f_c doit donc être inférieure à 1MHz.

III.4. Le bruit de phase

III.4.1. Principe

Conformément à la théorie développée précédemment (cf. §II.4.1) cette modélisation repose sur l'intégration d'un bruit blanc et d'un bruit de Flicker. Nous rappelons que, d'un point de vue fréquentiel (en utilisant la représentation de Laplace), une intégrale s'écrit :

$$\underbrace{y(t) = \int_0^t x(u).du}_{\text{domaine temporel}} \Leftrightarrow \underbrace{Y(p) = \frac{X(p)}{p}}_{\text{domaine fréquentiel}}
 \tag{Eq. 2 - 64}$$

$Y(p)$ et $X(p)$ sont les Transformées de Fourier de $y(t)$ et $x(t)$.

Comme les signaux sont discrets l'intervalle d'intégration est découpé en sous intervalles $[t_{n-1}, t_n[$.

Nous approximerons donc l'intégrale par la transformée bilinéaire $p = \frac{2}{T_{ech}} \cdot \frac{1-z^{-1}}{1+z^{-1}}$ soit le

développement suivant :

$$\phi(t) = \int_b \frac{\partial \phi(u)}{\partial u} \cdot du = \phi(t_{n-1}) + \left(\frac{\partial \phi(t_n)}{\partial t} + \frac{\partial \phi(t_{n-1})}{\partial t} \right) \cdot \frac{(t - t_{n-1})}{2} \text{ pour } t \in [t_{n-1}, t_n[\quad \text{Eq. 2 - 65}$$

Nous rappelons que (cf. Eq. 2 - 41) :

$$S_{\phi}(f)_{dBv/Hz} = S_{\partial \phi / \partial t}(f)_{dBv/Hz} - 20 \cdot \log(2 \cdot \pi \cdot f) \quad \text{Eq. 2 - 66}$$

III.4.2. Modèle

Pour intégrer plus aisément la fonctionnalité « bruit de phase (en $\frac{1}{f^2}$ et $\frac{1}{f^3}$) », nous avons défini une fonction génératrice réalisant l'intégration. Ce modèle est décrit ci-dessous :

```

FUNCTION f2_f3_noise (Fech, PSD_dBv_Hz_phase, fc :REAL; Beta_2
:INTEGER := 2 ; int: REAL_VECTOR(0 TO 17)) RETURN REAL_VECTOR IS

  CONSTANT Tsmpl      : REAL := 1.0/Fech;

  CONSTANT PSD_flicker: REAL :=
    PSD_dBv_Hz_phase+20.0*LOG10(MATH_2_PI* fc) ;

  VARIABLE outt      : REAL_VECTOR (0 TO 17);

  BEGIN

  outt(1 TO 17) := white_flicker_noise (Fech, PSD_dBv_Hz_flicker, fc, Beta_2,
int(1 TO 17));

  -- Phi_n := Phi_n_1 + (dPhi_n+dPhi_n_1)*Tsmpl/2 --
  outt(0) := int(0) + (outt(1)+int(1))*(Tsmpl/2.0);

  RETURN (outt);
END FUNCTION;
```

L'utilisateur entre la fréquence d'échantillonnage, la valeur de f_c , $S_{f2_f3}(f_c)$ β ainsi qu'un vecteur à 18 composantes.

Eq. 2 - 66

Nous utilisons la fonction *white_flicker_noise* qui est composée des fonctions *white_noise* (§III.2) et *flicker_noise* (§ III.3)

Eq. 2 - 65

La fonction retourne *outt(0)* dont la DSP est égale à : $S_{f2_f3}(f_c)$ à f_c ainsi que l'ensemble des états intermédiaires *outt(1 TO 17)*

Code 2 - 6 : Modèle en VHDL-AMS de la fonction génératrice de bruit en $\frac{1}{f^2}$ et $\frac{1}{f^3}$

III.4.3. Performances

Nous entrons comme paramètres : $S_{\frac{1}{f^2}}(f_c)_{dBv/Hz} = S_{\frac{1}{f^3}}(f_c)_{dBv/Hz} = -180dB_v/Hz$ et $f_c = 100kHz$

et nous effectuons ensuite une simulation temporelle pour extraire $N_{ech} = 2^{20}$ échantillons à la

fréquence d'échantillonnage $F_{ech} = 20MHz$. Le traitement fréquentiel s'effectue de nouveau sous SCILAB (cf. §II.1.3). Le périodogramme utilise des sous-fenêtres de 2^{14} échantillons. Nous utilisons une fenêtre Blackman7 pour éviter les phénomènes de saturation liés à une très grande composante continue, la bande de bruit équivalente sera $BBE = 3212.8Hz$.

Nous obtenons alors la figure suivante :

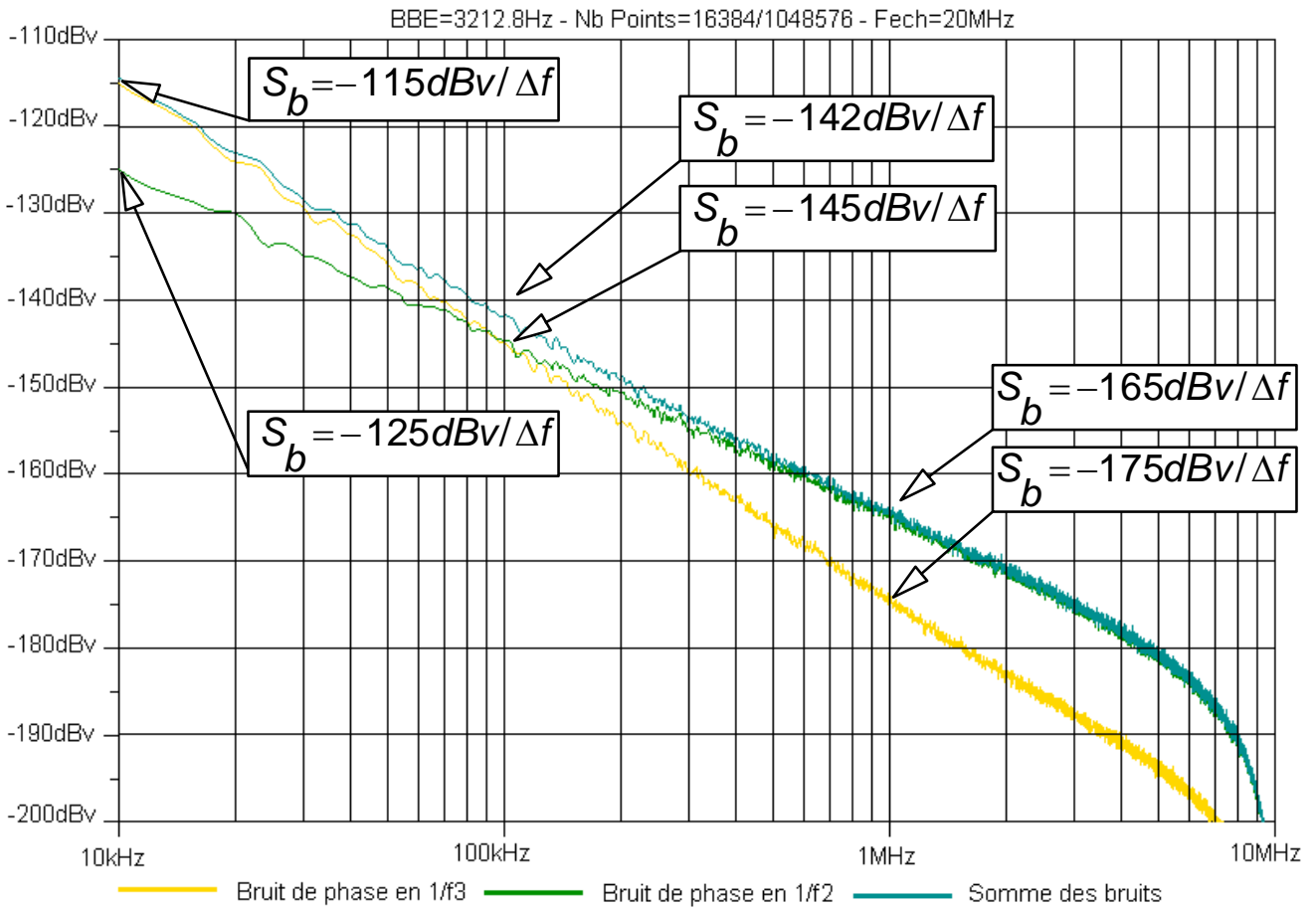


Figure 2 - 24 : Spectre du bruit de phase

N.B. : Comme précédemment afin de mieux appréhender la pente des DSP de bruit nous utilisons l'échelle log pour l'axe des fréquences, ce qui nous contraint à réduire l'intervalle de représentation. Ceci est sans conséquence sur les valeurs des DSP. La courbe complète en échelle de fréquence linéaire est disponible en Annexe 5 (cf. §II.5.2)

La pente du bruit de phase est alors de :

$$S_{\frac{1}{f^2}}(10kHz)_{dBv/Hz} - S_{\frac{1}{f^2}}(1MHz)_{dBv/Hz} \approx -40 \text{ dB}/2.\text{décades}$$

$$S_{\frac{1}{f^3}}(10kHz)_{dBv/Hz} - S_{\frac{1}{f^3}}(1MHz)_{dBv/Hz} \approx -60 \text{ dB}/2.\text{décades}$$

Eq. 2 - 67

Nous mesurons d'après le spectre les caractéristiques suivantes :

$$\begin{aligned}
 S_{\text{bruit}}(f)_{dB_v/Hz} &= S_{\text{bruit mesuré}}(f)_{dB_v/\Delta f} - 10 \cdot \log(3212.8) \\
 S_{\frac{1}{f^2}}(10\text{kHz})_{dB_v/Hz} &\approx -160\text{dB}_v/Hz \quad // \quad S_{\frac{1}{f^2}}(1\text{MHz})_{dB_v/Hz} \approx -200\text{dB}_v/Hz \\
 S_{\frac{1}{f^3}}(10\text{kHz})_{dB_v/Hz} &\approx -150\text{dB}_v/Hz \quad // \quad S_{\frac{1}{f^3}}(1\text{MHz})_{dB_v/Hz} \approx -210\text{dB}_v/Hz \\
 S_{\frac{1}{f^2}}(100\text{kHz})_{dB_v/Hz} &= S_{\frac{1}{f^3}}(100\text{kHz})_{dB_v/Hz} \approx -180\text{dB}_v/Hz \\
 S_{\text{somme}}(10\text{kHz})_{dB_v/Hz} &\approx -177\text{dB}_v/Hz
 \end{aligned}
 \tag{Eq. 2 - 68}$$

Les pentes et la fréquence f_c mesurées reprennent bien les paramètres appliqués au modèle.

La DSP de la somme des bruits en $\frac{1}{f^2}$ et $\frac{1}{f^3}$, notée $S_{b.somme}(f)$, suit la variation suivante :

$$\begin{aligned}
 S_{\text{somme}}(f) &\approx S_{\frac{1}{f^2}}(f) \quad \text{pour } f \ll f_c \\
 S_{\text{somme}}(f) &\approx S_{\frac{1}{f^3}}(f) \quad \text{pour } f \gg f_c \\
 S_{\text{somme}}(f) &\approx \begin{cases} S_{\frac{1}{f^2}}(f_c) \\ S_{\frac{1}{f^3}}(f_c) \end{cases} + 10 \cdot \log(2)
 \end{aligned}
 \tag{Eq. 2 - 69}$$

N.B. : On constate une chute de la DSP de bruit autour de $\frac{F_{ech}}{2}$. Cette chute est due à la transformation bilinéaire $p \rightarrow z$. En effet, si l'on remplace z par son expression fréquentielle, $z = e^{j \cdot 2 \cdot \pi \cdot f \cdot T_{ech}}$, on trouve :

$$\begin{aligned}
 \frac{1}{p} &= \frac{T_{ech}}{2} \cdot \frac{1+z^{-1}}{1-z^{-1}} = \frac{T_{ech}}{2} \cdot \frac{1+e^{-j \cdot 2 \cdot \pi \cdot f \cdot T_{ech}}}{1-e^{-j \cdot 2 \cdot \pi \cdot f \cdot T_{ech}}} \\
 &= \frac{T_{ech}}{2 \cdot j} \cdot \cotan\left(2 \cdot \pi \cdot f \cdot \frac{T_{ech}}{2}\right)
 \end{aligned}
 \tag{Eq. 2 - 70}$$

Soit :

$$\lim_{f \rightarrow \frac{F_{ech}}{2}} \log\left(\frac{T_{ech}}{2 \cdot j} \cdot \cotan\left(2 \cdot \pi \cdot f \cdot \frac{T_{ech}}{2}\right)\right) \rightarrow -\infty
 \tag{Eq. 2 - 71}$$

IV. MODELISATION DES GENERATEURS DE SIGNAUX

IV.1. Le générateur de sinusoïdes

IV.1.1. Principe

Ce modèle utilise les modules de bruit blanc (§III.2), de Flicker (§III.3) et de bruit de phase (§III.4) précédemment développés. Le générateur de sinusoïdes a l'équation suivante :

$$x(t) = A.(1 + b_A(t)).\sin(2.\pi.f_o.t + \Delta\phi_x(t)) \tag{Eq. 2 - 72}$$

Comme décrit ci-dessous, la variation d'amplitude $b_A(t)$ est composée d'un bruit blanc et d'un bruit de Flicker ($1/f$) et est définie par le couple $(f_{c_A}; S_{b_A(\text{blanc})}(f_{c_A})_{dBv/Hz})$. La variation de phase $\Delta\phi_x(t)$ a un comportement fréquentiel en $1/f^2$ et $1/f^3$ et est définie par le couple $(f_{c_{\Delta\phi_x}}; S_{\Delta\phi_x(1/f^2)}(f_{c_{\Delta\phi_x}})_{dBv/Hz})$. Concrètement, les bruit issus de $b_A(t)$ et $\Delta\phi_x(t)$ se combinent et équivalent à un bruit « d'amplitude » de $x(t)$. La DSP en dB_c/Hz du signal $x(t)$ sera notée : $S_x(f)_{dB_c/Hz}$.

Afin de faciliter l'utilisation du modèle, trois fréquences d'intersection $f_c > f'_c > f''_c$ ont été introduites à partir du spectre de $x(t)$:

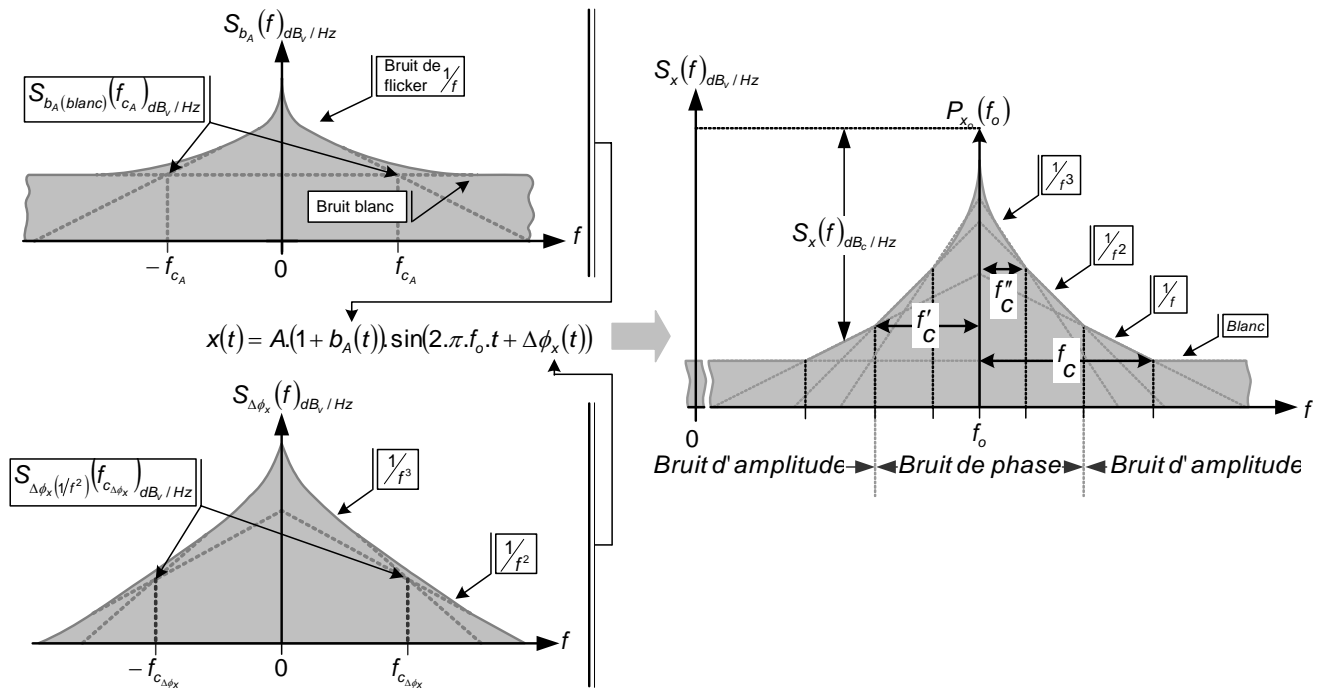


Figure 2 - 25 : Spectre du bruit de la sinusoïde

$S_x(f)_{dBc/Hz}$ est relié à $S_{b_A}(f)_{dBv/Hz}$ et $S_{\Delta\phi_x}(f)_{dBv/Hz}$ par (cf. §II.3.2 et §II.4.2) :

$$\begin{aligned} S_x(\pm f_o + f)_{dBc/Hz} &\approx S_{\Delta\phi_x} \frac{1}{f^3} (f)_{dBv/Hz} && \text{pour } -f'' < f < f'' \\ S_x(\pm f_o + f)_{dBc/Hz} &\approx S_{\Delta\phi_x} \frac{1}{f^2} (f)_{dBv/Hz} && \text{pour } -f'_c < f < -f''_c \text{ et } f'_c < f < f''_c \\ S_x(\pm f_o + f)_{dBc/Hz} &\approx S_{b_A} \frac{1}{f} (f)_{dBv/Hz} && \text{pour } -f_c < f < -f'_c \text{ et } f'_c < f < f_c \\ S_x(\pm f_o + f)_{dBc/Hz} &\approx S_{b_A \text{ blanc}}(f_{c_A})_{dBv/Hz} + 10.\log(2) && \text{pour } 0 < f < -f_c \text{ et } f_c < f \end{aligned} \quad \text{Eq. 2 - 73}$$

Nous en déduisons :

$$\begin{aligned} S_x(f_o \pm f'_c)_{dBc/Hz} &\approx S_{\Delta\phi_x}(f''_c)_{dBv/Hz} - 20.\log\left(\frac{f'_c}{f''_c}\right) && \text{avec } f_{c_{\Delta\phi_x}} = f''_c \\ S_x(f_o \pm f'_c)_{dBc/Hz} &\approx S_{b_A}(f_c)_{dBv/Hz} + 10.\log\left(\frac{f_c}{f'_c}\right) + 10.\log(2) && \text{avec } \frac{f_{c_A}}{2} = f_c \end{aligned} \quad \text{Eq. 2 - 74}$$

$$\text{Soit } \Rightarrow S_{b_A}(f_{c_A})_{dBv/Hz} = S_{\Delta\phi_x}(f_{c_x})_{dBv/Hz} - 20.\log\left(\frac{f'_c}{f_{c_{\Delta\phi_x}}}\right) - 10.\log\left(\frac{f_{c_A}}{f'_c}\right)$$

L'utilisateur entre les paramètres suivants : les trois fréquences d'offset (f_c, f'_c, f''_c) et $S_x(\pm f_o \pm f'_c)_{dBc/Hz}$. Le modèle calcule ensuite en interne la valeur des couples $(f_{c_A}; S_{b_A(\text{blanc})}(f_{c_A})_{dBv/Hz})$ et $(f_{c_{\Delta\phi_x}}; S_{\Delta\phi_x}(1/f^2)(f_{c_{\Delta\phi_x}})_{dBv/Hz})$.

IV.1.2. Modèle

Bien que notre source sinusoïdale soit une source analogique temps continu, les sources de bruit sont, elles, discrètes. Nous devons donc introduire une fréquence de génération du bruit que l'on prendra égale à la fréquence d'échantillonnage.

ENTITY analog_random_sinus IS

```
GENERIC (
    frequency           : REAL := 1.0e+5;
    amplitude           : REAL := SQRT(2.0*10.0**(20.0/10.0));
    Phase               : REAL := 0.0;
    PSD_dBc_Hz_sinus   : REAL := -160.0;
    fc_phase            : REAL := 1.0e+4;
    fc_flicker          : REAL := 5.0e+4;
    fc_amp              : REAL := 1.0e+5;
    Fech                : REAL := 2.0e+6);
```

PORT (TERMINAL Psinp, Psinn : electrical);
END ENTITY analog_random_sinus;

ARCHITECTURE behaviour OF analog_random_sinus IS

```
CONSTANT fc_amplitude      : REAL := fc_amp*2.0;
CONSTANT PSD_dBv_Hz_phase : REAL := PSD_dBc_Hz_sinus ;
CONSTANT PSD_dBv_Hz_amp    : REAL := PSD_dBc_Hz_phase -
    20.0*LOG10(fc_flicker/ fc_phase)-10.0*LOG10(fc_amplitude/ fc_flicker);
```

QUANTITY Vout ACROSS Iout THROUGH Psinp TO Psinn;

L'utilisateur définit les paramètres :

- fréquence
- amplitude
- phase à l'origine
- $S_x(\pm f_o \pm f'_c)_{dBc/Hz}$
- f''_c, f'_c, f_c
- F_{ech}

Calcul de f_{c_A}

Calcul de $S_{\phi_x}(f_{c_x})_{dBv/Hz}$

Calcul de $S_{b_A}(f_{c_A})_{dBv/Hz}$

<pre> SIGNAL Phi, Amp : REAL := 0.0 ; BEGIN Amp_phase_generation : PROCESS VARIABLE rand_amp : REAL_VECTOR (0 TO 16) := (I.C); VARIABLE rand_phase : REAL_VECTOR (0 TO 17) := (I.C); BEGIN rand_phase := f2_f3_noise (Fech,PSD_dBv_Hz_phase,fc_phase,2,rand_phase); Phi <= rand_phase(0); rand_amp:=white_flicker_noise(Fech,PSD_dBv_Hz_amplitude,fc_amplitude,2, rand_amp) ; Amp <= amplitude(1.0 + rand_amp(0)); WAIT FOR 1.0/Fech; END PROCESS Amp_phase_generation; BREAK ON (Phi,Amp); Vout == Amp*sin(MATH_2_PI*frequency*NOW + Phi + Phase); END ARCHITECTURE behaviour; </pre>	<p>Début du PROCESS I.C. : Conditions initiales</p> <p>Appel (III.4) de la fonction <i>f2_f3_noise</i>, pour le bruit de phase $\Delta\phi_x$</p> <p>Appel de la fonction <i>white_flicker_noise</i> composée des fonctions (§III.2, III.3) pour le bruit d'amplitude b_A</p> <p>Répétition du PROCESS tous les 1/Fech</p> <p>Eq. 2 - 72</p>
--	--

Code 2 - 7 : Modèle en VHDL-AMS d'un générateur de sinusoides

IV.1.3. Performances

Pour mesurer les performances de notre modèle nous avons de nouveau recours aux fonctions de FFT développées par M. ROBBE sous SCILAB (§II.1.3). Les conditions d'utilisation du périodogramme sont les suivantes : $F_{ech} = 2MHz$, 2^{20} échantillons décomposé en sous-fenêtres de 2^{15} échantillons. Nous utilisons une fenêtre de Blackman7 ce qui induit une bande de bruit équivalente : $BBE = 160.6Hz$.

Les paramètres suivants sont appliqué au modèle : $f_o = 500kHz$, $A \approx 14.14V$ soit une puissance de $20dB_v$ (répartie sur deux raies de $17dB_v$ chacune), $F_{ech} = 2MHz$, la DSP de la sinusoïde $S_x(\pm f_o \pm f_c)_{dB_c/Hz} = -160dB_c/Hz$ et les fréquences d'intersection (ou d'offset) : $f_c'' = 10kHz$, $f_c' = 50kHz$, $f_c = 100kHz$.

Nous espérons trouver théoriquement du point de vue de la porteuse :

$$\begin{aligned}
 S_x(\pm 500 \pm 10kHz)_{dB_c/Hz} &= -160dB_c/Hz \\
 S_x(\pm 500 \pm 50kHz)_{dB_c/Hz} &= -160 - 20 \cdot \log\left(\frac{5e^4}{1e^4}\right) = -174dB_c/Hz \\
 S_x(\pm 500 \pm 100kHz)_{dB_c/Hz} &= -160 - 20 \cdot \log\left(\frac{5e^4}{1e^4}\right) - 10 \cdot \log\left(\frac{1e^5}{5e^4}\right) = -177dB_c/Hz
 \end{aligned}$$

Eq. 2 - 75

Ce qui équivaut à un bruit d'amplitude et de phase de DSP :

$$S_{\Delta\phi_x}(f_{c_{\Delta\phi_x}} = f_c'')_{dB_v / Hz} = -160 dB_v / Hz$$

$$S_{b_A}(f_{c_A} = 2.f_c)_{dB_v / Hz} = -160 - 20 \cdot \log\left(\frac{5e^4}{1e^4}\right) - 10 \cdot \log\left(\frac{2e^5}{5e^4}\right) = -180 dB_v / Hz \quad \text{Eq. 2 - 76}$$

Afin d'avoir une bonne lisibilité des courbes, nous réduisons notre domaine de visualisation à l'intervalle [300kHz,700kHz]. La courbe complète est disponible en Annexe 5 (cf. §II.5.3).

Après simulation et traitement des données, nous obtenons la figure suivante :

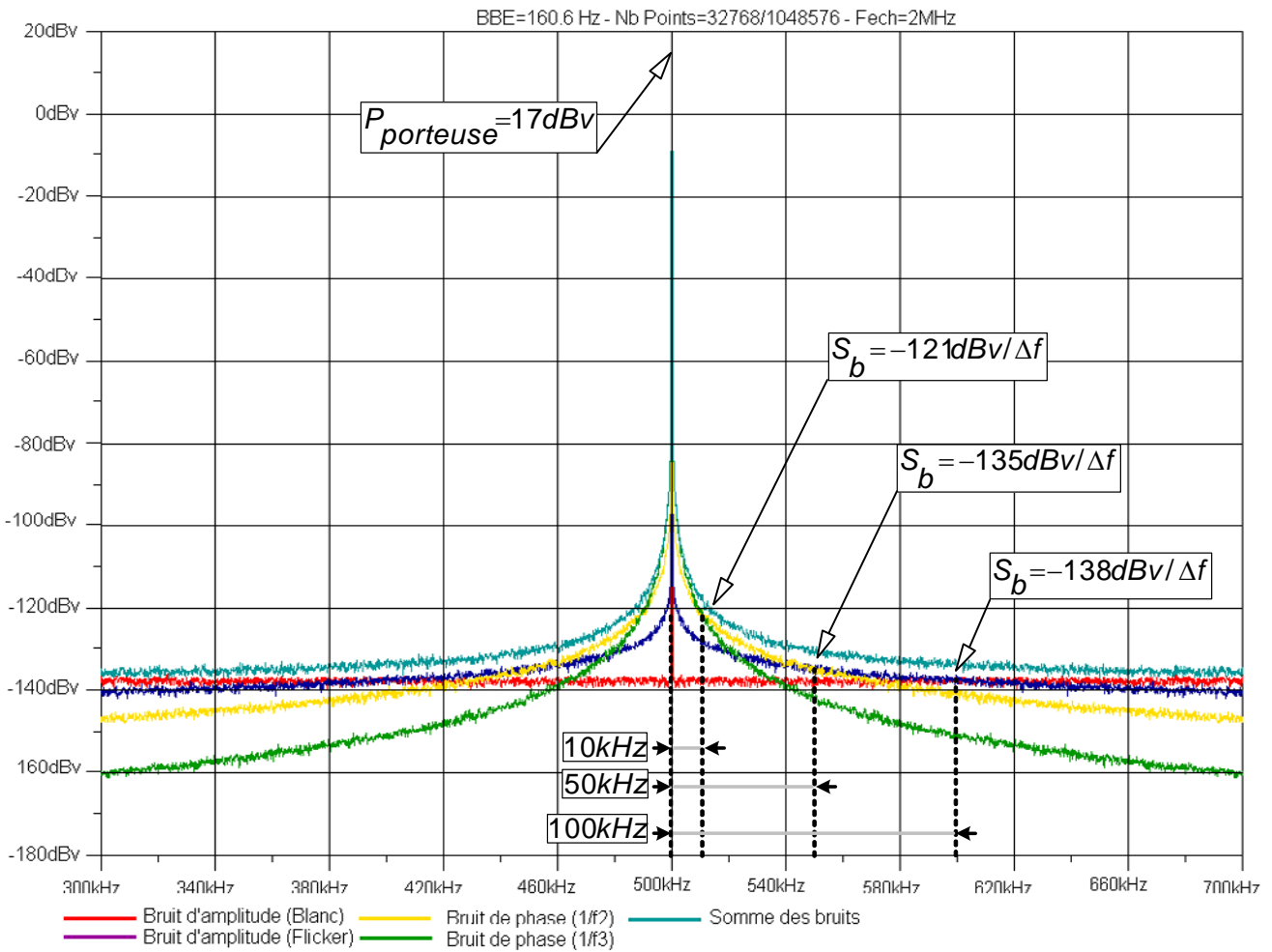


Figure 2 - 26 : Spectre du bruit de phase du générateur de sinusoïdes

Soit une DSP égale à :

$$S_{bruit}(f)_{dB_c / Hz} = S_{bruit\ mesur\acute{e}}(f)_{dB_v / Hz} - P_{porteuse} - 10 \cdot \log(160.6)$$

$$S_{x \frac{1}{f^3}}(\pm 500 \pm 10kHz)_{dB_c / Hz} = S_{x \frac{1}{f^2}}(\pm 500 \pm 10kHz)_{dB_c / Hz} \approx -160 dB_c / Hz$$

$$S_{x \frac{1}{f^2}}(\pm 500 \pm 50kHz)_{dB_c / Hz} = S_{x \frac{1}{f}}(\pm 500 \pm 50kHz)_{dB_c / Hz} \approx -174 dB_c / Hz \quad \text{Eq. 2 - 77}$$

$$S_{x \frac{1}{f}}(\pm 500 \pm 100kHz)_{dB_c / Hz} = S_{x\ blanc}(\pm 500 \pm 100kHz)_{dB_c / Hz} \approx -177 dB_c / Hz$$

Les mesures des fréquences d'intersection et des valeurs de la DSP du modèle sont conformes à nos attentes, le modèle est donc validé.

IV.2. Le générateur d'horloge

IV.2.1. Principe

Ce modèle utilise également les modules de bruit blanc et de Flicker (§III.2, III.3) et de bruit de phase (§III.4) précédemment développés. Il intègre deux types de bruit :

le jitter PM défini par une variation aléatoire de l'instant d'échantillonnage, $t'_n = T_{ech}(n + b_{PM}(n))$ conformément à la théorie développée (§II.3.3, §II.3.4)

le jitter FM défini par une variation aléatoire de l'instant d'échantillonnage, défini par $t''_n = T_{ech}\left(n - \frac{\Delta\phi_{FM}(t''_n)}{2.\pi}\right)$ conformément à la théorie développée (§II.4.3)

Si l'oscillateur est soumis aux deux types de jitter, nous constatons qu'à PSD égales, le jitter PM et le jitter FM n'ont pas le même impact sur l'instant d'échantillonnage, puisque le jitter FM est divisé par $2.\pi$.

Comme décrit ci-dessous, le jitter PM $b_{PM}(t)$ est composé d'un bruit blanc et d'un bruit de Flicker ($1/f$) et est défini par le couple $(f_{c_{PM}}; S_{b_{PM}(blanc)}(f_{c_{PM}})_{dBv/Hz})$. Le jitter FM $\Delta\phi_{FM}(t)$ a une variation fréquentielle en $1/f^2$ et $1/f^3$ et est défini par le couple $(f_{c_{FM}}; S_{\Delta\phi_{FM}(1/f^2)}(f_{c_{FM}})_{dBv/Hz})$. Le jitter FM précède dans le domaine fréquentiel le jitter PM.

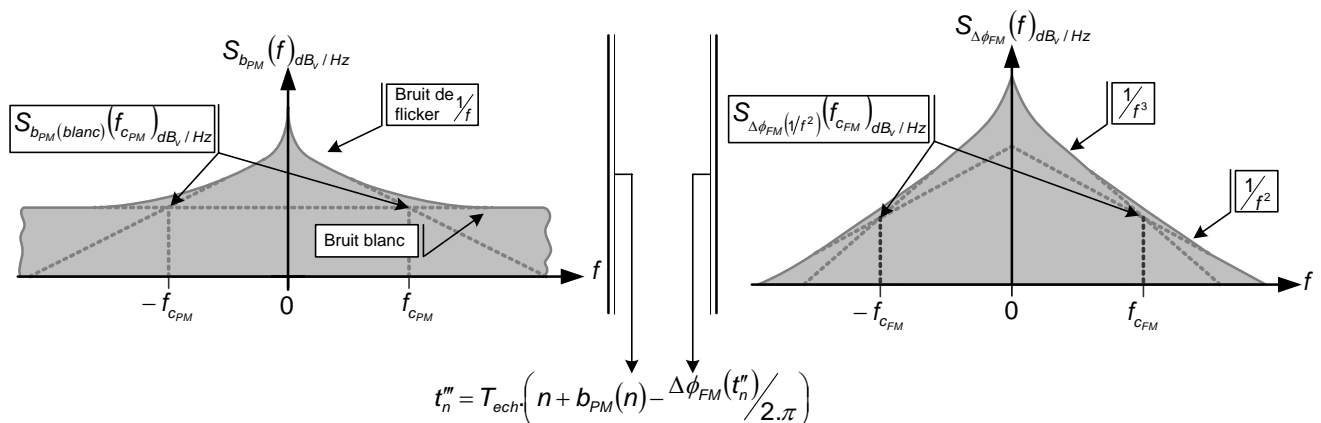


Figure 2 - 27 : Composition du jitter (FM+PM)

L'utilisateur du modèle doit alors entrer les paramètres qui définissent les deux types de jitter :

$$(f_{c_{PM}}; S_{b_{PM}(blanc)}(f_{c_{PM}})_{dBv/Hz}) \text{ et } (f_{c_{FM}}; S_{\Delta\phi_{FM}(1/f^2)}(f_{c_{FM}})_{dBv/Hz}).$$

IV.2.2. Modèle

Les deux fronts sont indépendants l'un de l'autre et la répartition est décrite ci-dessous :

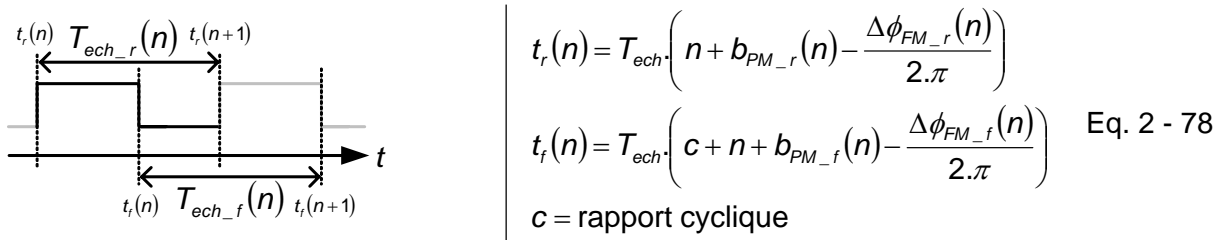


Figure 2 - 28 : Répartition des fronts d'horloge

N.B. : Les fronts montants et descendants sont indépendants, mais sont modélisés suivant le même principe. Afin de ne pas surcharger l'explication du modèle, nous ne traiterons que le front montant.

Le modèle est explicité ci-dessous :

```

-----
ENTITY digital_random_pulse IS
GENERIC (
    frequency          : REAL := 2.0e+6 ;
    delay              : REAL := 0.0 ;
    c_r                : REAL := 0.5 ;
    PSD_dBv_Hz_FM     : REAL := -148.0 ;
    fc_FM              : REAL := 1.0e+4 ;
    PSD_dBv_Hz_PM     : REAL := -184.0 ;
    fc_PM              : REAL := 2.0e+5 ;
    PORT (Npulse: INOUT BIT := '0');
END ENTITY digital_random_pulse;
-----
ARCHITECTURE behaviour OF digital_random_pulse IS

SIGNAL: FM, PM: REAL := 0.0 ;

BEGIN

Output_generation : PROCESS
VARIABLE period_r_int, period_f_int : REAL := 1.0/frequency;
VARIABLE rand_r_edge, rand_f_edge   : REAL_VECTOR (0 TO 16) := (I.C.);
VARIABLE rand_phase_r, rand_phase_f : REAL_VECTOR (0 TO 17) := (I.C.);
VARIABLE edge_r, edge_f             : REAL := 0.0;
VARIABLE Phi_r, Phi_f               : REAL := 0.0;
VARIABLE Waiting_time               : REAL := 0.0;

BEGIN

IF (NOW = 0.0) AND (delay /= 0.0) THEN WAIT FOR delay;
ELSE IF Npulse = '0' THEN
----- FM jitter rising edge -----
rand_phase_f := f2_f3_noise (1.0/period_f_int, PSD_dBv_Hz_FM, fc_FM_jitter,2
, rand_phase_f);

ASSERT ((1.0 + rand_phase_f(0)-Phi_f/MATH_2_PI)/frequency > 0.0)
REPORT "NEGATIV PERIOD UP" SEVERITY warning;

IF (1.0 + rand_phase_f(0)-Phi_f/MATH_2_PI)/frequency > 0.0 THEN

```

L'utilisateur définit les paramètres :

- fréquence
- délai initial
- rapport cyclique
- $S_{\Delta\phi_{FM}}(f_{c_{FM}})$ dBV / Hz
- $f_{c_{FM}}$
- $S_{b_{PM}}(f_{c_{PM}})$ dBV / Hz
- $f_{c_{PM}}$

Début du PROCESS
I.C. : Conditions initiales

Délais initial
Si front montant

Appel (III.4) de la fonction f2_f3_noise, pour du jitter FM $\Delta\phi_{FM}$

Vérification de la positivité de la nouvelle période

```

period_f_int := ( 1.0 + (rand_phase_f(0)-Phi_f/MATH_2_PI))/frequency;
Phi_f       := rand_phase_f(0);
FM_jitter   <= Phi_f;
END IF;
----- PM jitter rising edge -----
rand_f_edge := white_flicker_noise (1.0/period_f_int, PSD_dBv_Hz_PM,
                                   fc_PM_jitter, 2,rand_f_edge);

edge_f      := rand_f_edge(0);
PM_jitter   <= edge_f;
ASSERT ((c_r+(Phi_f-Phi_r)/MATH_2_PI +edge_f-edge_r)/frequency>0.0)
REPORT "NEGATIV PULSE WIDTH UP" SEVERITY warning;

IF(c_r+(Phi_f-Phi_r)/MATH_2_PI+edge_f-edge_r)/frequency>0.0 THEN
waiting_time := (c_r+(Phi_f-Phi_r)/MATH_2_PI+edge_f-edge_r)/frequency;
END IF;
Npulse <= '1'; WAIT FOR waiting_time;

{ELSE
----- FM jitter falling edge -----
----- PM jitter falling edge -----
Npulse <= '0'; WAIT FOR waiting_time;}

END IF;
END PROCESS Output_generation;
END ARCHITECTURE behaviour;

```

Appel de la fonction white_flicker_noise composée des fonctions (§III.2, III.3) pour le jitter PM b_{PM}

Vérification de la positivité de la durée de l'impulsion haute Eq. 2 - 78

Même processus pour le front descendant

Code 2 - 8 : Modèle en VHDL-AMS d'un générateur d'horloge bruitée par un jitter FM et PM

IV.2.3. Performances

Pour mesurer les performances de notre horloge bruitée, nous avons recours à l'échantillonnage d'une porteuse idéale $x_o(t) = A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t)$. La mesure de ce signal échantillonné et l'utilisation des formules Eq. 2 - 36 et Eq. 2 - 52 nous permettent ensuite de remonter jusqu'aux caractéristiques de l'horloge.

Les performances sont toujours évaluées avec les fonctions de Fourier développées par M. ROBBE sous SCILAB (§II.1.3). Les conditions d'utilisation sont identiques à la précédente simulation. Nous entrons comme paramètres : $f_o = 500kHz$, $A \approx 14.14V$ soit une puissance de $20dB_V$ (répartie sur deux raies de $17dB_V$ chacune), $F_{ech} = 2MHz$, $f_{c_{FM}} = 10kHz$ $S_{\Delta\phi_{FM}}(f_{c_{FM}})_{dB_V/Hz} = -148dB_V/Hz$, $f_{c_{PM}} = 200kHz$ et $S_{b_{PM}}(f_{c_{PM}})_{dB_V/Hz} = -184dB_V/Hz$.

Comme pour la partie précédente, pour avoir une bonne lisibilité des courbes nous réduisons notre domaine de visualisation à l'intervalle $[300kHz, 700kHz]$, néanmoins la courbe complète est disponible en Annexe 5 (cf. §II.5.4)

Nous obtenons, après traitement, la figure suivante :

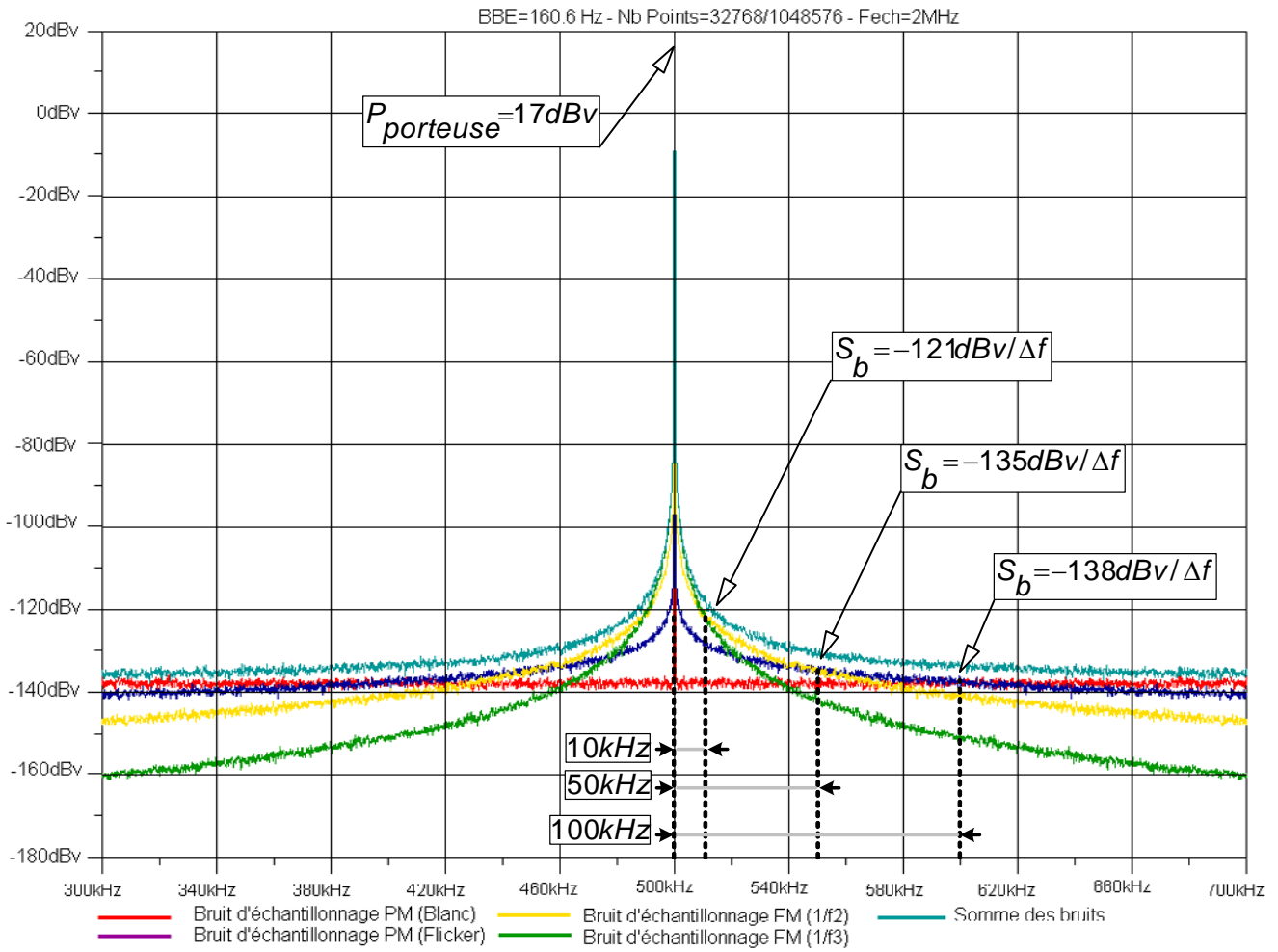


Figure 2 - 29 : Spectre du bruit de phase d'un échantillonnage jitté (FM et PM jitter)

Nous mesurons une DSP égale à :

$$S_{bruit}(f)_{dB_c/Hz} = S_{bruit\ mesur\acute{e}}(f)_{dB_v/Hz} - P_{porteuse} - 10 \cdot \log(160.6)$$

$$S_{ech_FM \frac{1}{f^3}}(\pm 500 \pm 10kHz)_{dB_c/Hz} = S_{ech_FM \frac{1}{f^2}}(\pm 500 \pm 10kHz)_{dB_c/Hz} \approx -160dB_c/Hz$$

$$S_{ech_FM \frac{1}{f^2}}(\pm 500 \pm 50kHz)_{dB_c/Hz} = S_{ech_PM \frac{1}{f}}(\pm 500 \pm 50kHz)_{dB_c/Hz} \approx -174dB_c/Hz \quad \text{Eq. 2 - 79}$$

$$S_{ech_PM \frac{1}{f}}(\pm 500 \pm 100kHz)_{dB_c/Hz} = S_{ech_PM\ blanc}(\pm 500 \pm 100kHz)_{dB_c/Hz} \approx -177dB_c/Hz$$

Soit pour les fréquences de coupure :

$$f_c = 100kHz; f'_c = 50kHz; f''_c = 10kHz \quad \text{Eq. 2 - 80}$$

N.B. : Le choix des valeurs de la DSP du jitter de phase n'est pas évidemment le fruit du hasard. Nous démontrons par la même occasion qu'un bruit mesuré peut avoir deux origines : soit le générateur de sinusoides est bruité, soit c'est l'horloge d'échantillonnage qui l'est.

Pour remonter aux DSP et fréquence de coupure des jitter FM et PM nous devons effectuer un rappel théorique des parties §II.3.4 et §II.4.3. Soit l'illustration suivante :

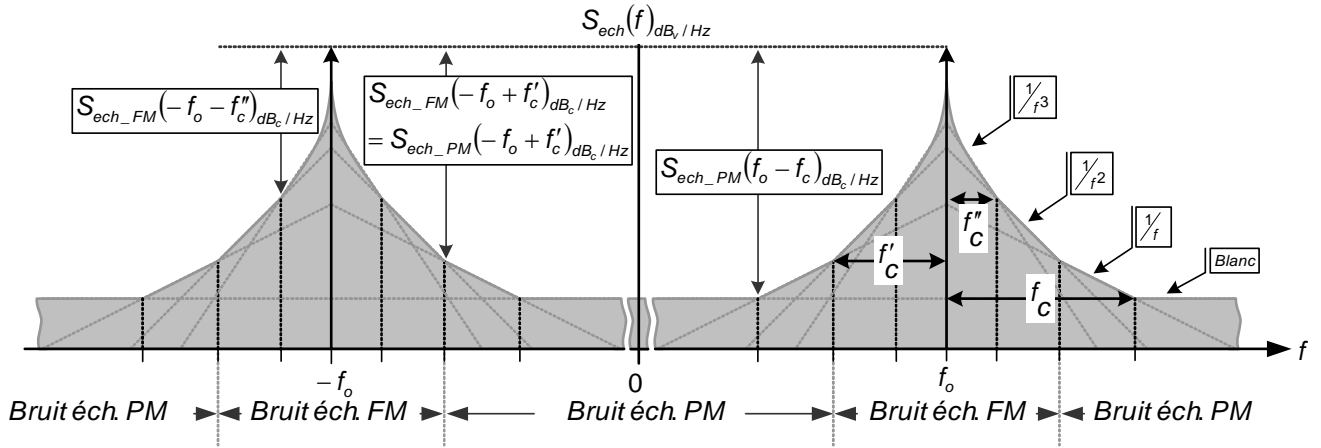


Figure 2 - 30 : Spectre théorique du bruit de phase de l'échantillonnage jitté d'une porteuse idéale

Le spectre d'échantillonnage de la porteuse par l'horloge jittée se répartit de la manière suivante :

$$\begin{aligned}
 S_{ech_FM}(\pm f_0 + f)_{dBc/Hz} &\approx 20 \cdot \log\left(\frac{f_0}{F_{ech}}\right) + S_{\Delta\phi_{FM}} \frac{1}{f^3}(f)_{dB_v/Hz} && \text{pour } -f''_c < f < f''_c \\
 S_{ech_FM}(\pm f_0 + f)_{dBc/Hz} &\approx 20 \cdot \log\left(\frac{f_0}{F_{ech}}\right) + S_{\Delta\phi_{FM}} \frac{1}{f^2}(f)_{dB_v/Hz} && \text{pour } \begin{cases} -f'_c < f < -f''_c \\ f''_c < f < f'_c \end{cases} \\
 S_{ech_PM}(\pm f_0 + f)_{dBc/Hz} &\approx 20 \cdot \log\left(\frac{2 \cdot \pi \cdot f_0}{F_{ech}}\right) + S_{b_{PM}} \frac{1}{f}(f)_{dB_v/Hz} && \text{pour } \begin{cases} -f_c < f < -f'_c \\ f'_c < f < f_c \end{cases} \\
 S_{ech_PM}(\pm f_0 + f)_{dBc/Hz} &\approx 20 \cdot \log\left(\frac{2 \cdot \sqrt{2} \cdot \pi \cdot f_0}{F_{ech}}\right) + S_{b_{PM} \text{ blanc}}(f_{c_{PM}})_{dB_v/Hz} && \text{pour } \begin{cases} 0 < f < -f_c \\ f_c < f \end{cases}
 \end{aligned} \tag{Eq. 2 - 81}$$

Avec les relations suivantes pour les fréquences de coupures :

$$f_c = f_{c_{PM}} / 2; f'_c = \frac{(f_{c_{FM}})^2}{f_{c_{PM}}} \cdot 10^{\frac{(S_{\phi_{FM}}(f_{c_{FM}})_{dB_v/Hz} - S_{b_{PM} \text{ blanc}}(f_{c_{PM}}) - 20 \cdot \log(2 \cdot \pi))}{10}}; f''_c = f_{c_{FM}} \tag{Eq. 2 - 82}$$

Ce qui correspond dans notre cas aux fréquences de coupures des jitter :

$$f_{c_{PM}} = 200kHz; f_{c_{FM}} = 10kHz \tag{Eq. 2 - 83}$$

Nous en déduisons donc les valeurs des DSP des jitter :

$$\begin{aligned}
 S_{\Delta\phi_{FM}} \left(\frac{1}{f^2}\right) (f_{c_{FM}})_{dB_v/Hz} &= \left\{ \begin{aligned} &-160 - 20 \cdot \log\left(\frac{5e^5}{2e^6}\right) \\ &-174 - 20 \cdot \log\left(\frac{5e^5}{2e^6}\right) - 20 \cdot \log\left(\frac{5e^4}{1e^4}\right) \end{aligned} \right\} = -148dB_v/Hz \\
 S_{b_{PM} \text{ (blanc)}} (f_{c_{PM}})_{dB_v/Hz} &= \left\{ \begin{aligned} &-174 - 20 \cdot \log\left(\frac{2 \cdot \pi \cdot 5e^5}{2e^6}\right) - 10 \cdot \log\left(\frac{2e^5}{5e^4}\right) \\ &-177 - 20 \cdot \log\left(\frac{2 \cdot \sqrt{2} \cdot \pi \cdot 5e^5}{2e^6}\right) \end{aligned} \right\} = -184dB_v/Hz
 \end{aligned} \tag{Eq. 2 - 84}$$

Ces mesures sont conformes à nos estimations théoriques, ce qui valide notre modèle.

V. CONCLUSION

Nous avons développé dans la première partie l'étude théorique des sources de bruit, afin d'en extraire les relations liant l'expression des sources de bruit dans le domaine temporel et leur expression dans le domaine fréquentiel. Nous avons ensuite, dans la seconde partie, mis en pratique ces relations en créant une bibliothèque de modèles comportementaux en langage VHDL-AMS, de sources de bruits caractérisées en temps discret. Une fois que les propriétés modèles de bruit blanc, de bruit de Flicker, de bruit en $\frac{1}{f^2}$ et de bruit en $\frac{1}{f^3}$ ont été vérifiées, nous avons vérifié la cohérence de nos modèles et les intégrant dans un générateur de sinusoïdes et un générateur d'horloge. L'étude du bruit de la sinusoïde d'un part et de l'échantillonnage d'une porteuse idéale par notre générateur d'horloge d'autre part nous ont permis de valider l'ensemble de nos modèles.

Nous pouvons donner un ordre d'idée de la vitesse de simulation par les exemples suivants :

les 5×2^{20} échantillons du générateur de sinusoïde bruitée utilisés pour obtenir la Figure 2 - 26 nécessitent 8'50 de temps CPU sur une station SunBlade 150 ;

les 5×2^{20} échantillons obtenus par l'échantillonnage jitté d'une porteuse idéale utilisés pour obtenir la Figure 2 - 29 nécessitent 19'50 de temps CPU sur une station SunBlade 150.

L'étape suivante de notre travail de modélisation, va consister à extraire les paramètres réels des sources de bruits afin de pouvoir retranscrire les conditions réelles d'utilisation de notre chaîne de réception RF. Nous allons principalement nous intéresser au modulateur Sigma Delta à temps discret, objet principal de l'étude de la thèse.

CHAPITRE III :
Mesure du jitter interne
d'une structure simple à capacités commutées

I. INTRODUCTION

Le but de ce chapitre est d'évaluer le jitter interne d'une structure utilisant le principe des capacités commutées. Pour cela ce chapitre est divisé en 3 parties.

La première partie a pour but la justification de notre démarche, à savoir prouver que les effets du jitter dans un modulateur Sigma Delta sont principalement localisés dans le filtre à capacités commutées. Nous pouvons donc concentrer notre étude sur une structure simple de test et ensuite extrapoler les résultats obtenus aux modulateurs qui nous intéressent.

La seconde partie décrit cette structure de test ainsi que le protocole de mesure qui va nous permettre, étape par étape, de remonter aux caractéristiques statistiques du jitter interne en partant des caractéristiques statistiques du bruit d'amplitude de la sortie du montage de mesure.

La troisième partie concerne le banc de mesure, les mesures et leur interprétation.

II. Localisation des effets du jitter interne

Comme nous l'avons exposé dans le Chapitre I, l'architecture d'un modulateur Sigma Delta est une structure bouclée, c'est-à-dire que le résultat à un instant donné, dépend des résultats des instants précédents. Considérons dans un premier temps un modulateur Sigma Delta à temps discret du premier ordre dont le schéma de principe est le suivant :

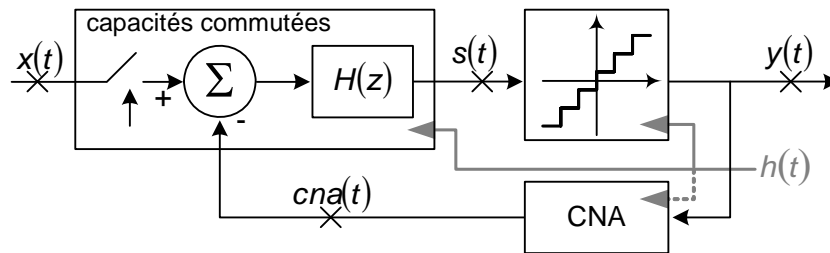


Figure 3 - 1 : Modulateur Sigma Delta du premier ordre

Dans notre cas d'étude, ce modulateur est composé d'un quantificateur, d'un Convertisseur Numérique/Analogique et d'une structure à capacités commutées qui remplit les fonctions d'échantillonnage du signal d'entrée, d'addition/soustraction et de filtrage. La structure à capacités commutées ainsi que le quantificateur sont cadencés par une horloge $h(t)$ de fréquence F_{ech} . Le CNA est également piloté, indirectement, par l'horloge puisque son signal d'entrée est délivré par le quantificateur qui lui est synchronisé.

Supposons maintenant que l'horloge $h(t)$ soit sujette à des variations aléatoires dans le domaine temporel. Les fluctuations temporelles peuvent être imputées au générateur d'horloge externe ou au générateur interne d'horloges non recouvrantes ou encore à toutes autres variations d'une tension de seuil retardant ou accélérant ainsi un évènement. Nous ne nous intéressons dans cette partie qu'au jitter interne au modulateur.

Ces fluctuations temporelles affecteront dans un tel cas l'ensemble de la boucle, entraînant une dégradation des performances globales du modulateur. Nous constatons cependant que les effets sur chacun des blocs ne sont pas identiques et que certains blocs contribuent plus que d'autres à cette dégradation.

Afin d'identifier le ou les blocs les plus sensibles aux effets du jitter, nous devons entrer un peu plus dans le détail du fonctionnement d'un modulateur et effectuer une étude temporelle de ce dernier. Pour ce faire nous prendrons comme exemple le modulateur Sigma Delta passe-bas du premier ordre précédemment décrit (cf. Chapitre I, Figure 1 – 44).

Nous avons ajouté au schéma les arbres d'horloges afin de pouvoir isoler le ou les blocs potentiellement sensibles au jitter.

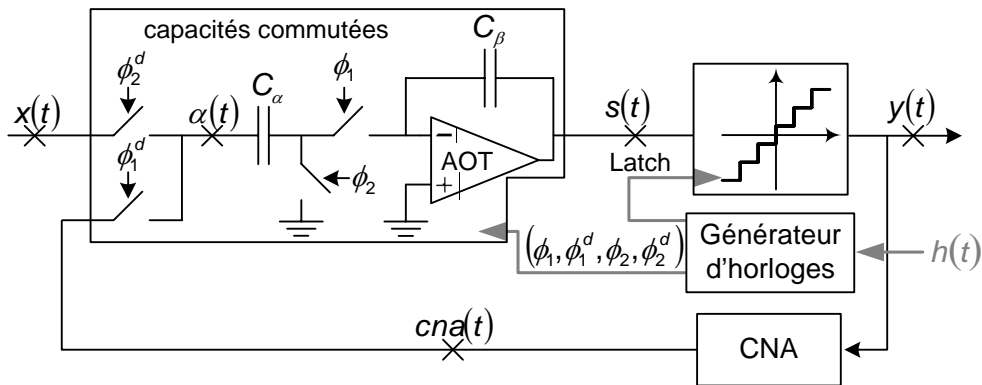


Figure 3 - 2 : Modulateur Sigma Delta passe bas du premier ordre

Pour faciliter la compréhension, l'Amplificateur Opérationnel à Transconductance ainsi que les interrupteurs sont considérés idéaux. De plus, dans un premier temps, les horloges de commande des interrupteurs CMOS seront exemptes de jitter. Nous détaillons ci-dessous les chronogrammes des entrées/sorties du modulateur Sigma Delta précédent :

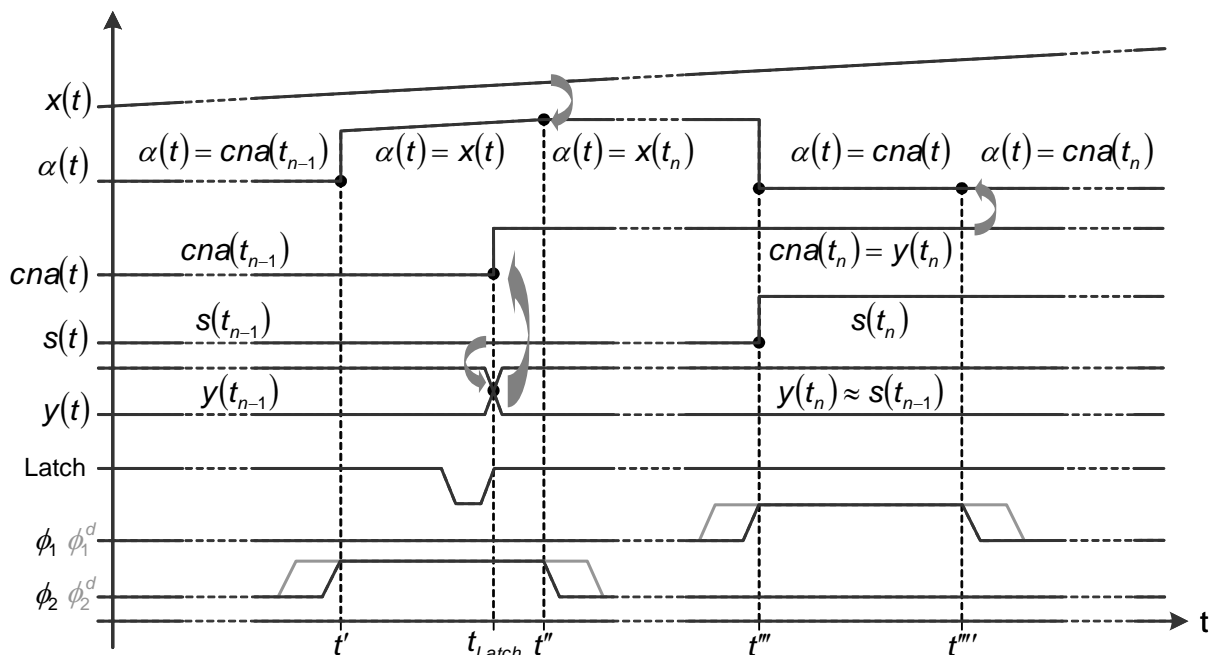


Figure 3 - 3 : Chronogramme d'un cycle du modulateur

$[t', t'']$ et $[t''', t''']$ correspondent aux phases de charge de la capacité C_α respectivement à la valeur de $x(t)$ puis de $cna(t)$. t_{Latch} correspond à l'instant d'échantillonnage et de conversion du CAN.

Comme nous l'avons démontré dans le Chapitre II, une variation aléatoire de l'instant d'échantillonnage (jitter d'horloge $\delta(t)$) d'une sinusoïde $x_o(t) = A \cdot \sin(2 \cdot \pi \cdot f_o \cdot t)$, induit un bruit d'amplitude proportionnelle à la pente du signal échantillonné (cf. Eq. 2 - 33 et Eq. 2 - 34).

Ainsi plus la pente du signal échantillonné sera importante et plus les effets du jitter d'horloge seront grands :

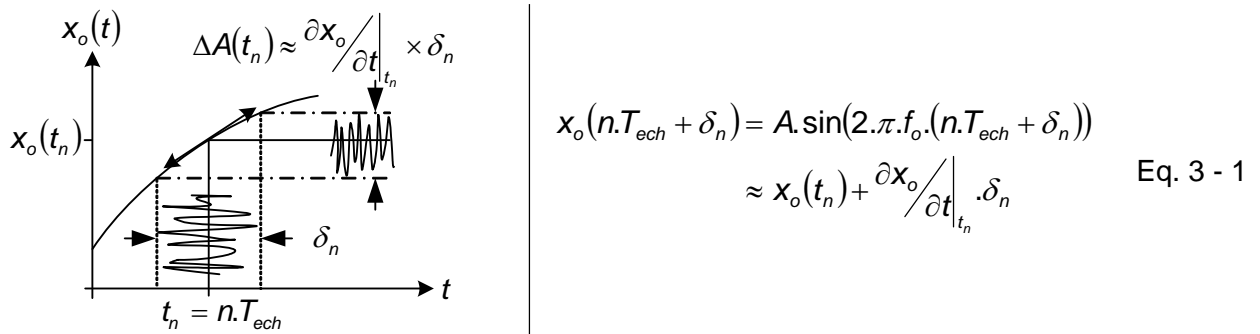


Figure 3 - 4 : Echantillonnage sujet à jitter

Or après étude du chronogramme, nous constatons que l'ensemble des signaux internes au modulateur ($\alpha(t), s(t), cna(t)$) sont maintenus constants lors de leur échantillonnage à l'exception du signal d'entrée $x(t)$. Nous en déduisons qu'il sera le plus affecté par le jitter. Ainsi un jitter sur les instants (t', t_{Latch}, t'', t''') n'apporte qu'une faible contribution au jitter global et seul le jitter sur l'instant d'échantillonnage t'' du signal $x(t)$ est prépondérant.

Dans ces conditions nous pouvons considérer que l'effet du jitter d'horloge est prépondérant dans le filtre et plus particulièrement dans l'interrupteur qui effectue l'échantillonnage, celui commandé par la phase ϕ_2 . Ainsi nous pouvons, en première approximation, modéliser un modulateur Sigma Delta dont l'horloge de commande est soumise à un jitter, par un modulateur idéal dont le signal d'entrée est issu d'un échantillonnage jitté :

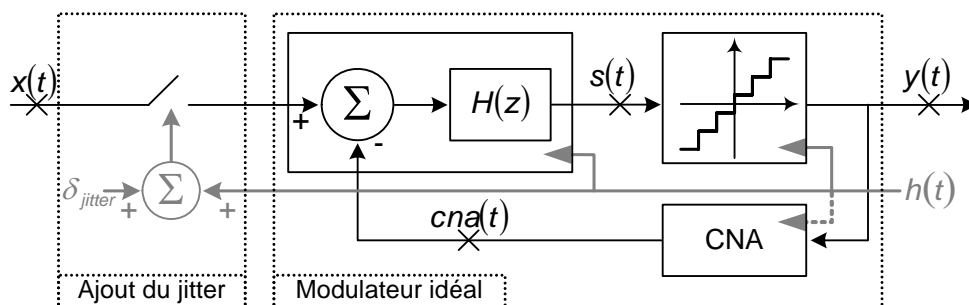


Figure 3 - 5 : Modélisation de l'effet du jitter dans un modulateur Sigma Delta temps discret

N.B. : nous avons déjà effectué une telle modélisation lors de l'étude des effets du jitter (PM et FM) en introduisant le signal $y_{FFT}(t)$ qui permettait de prendre en compte les variations d'amplitude sans les variations temporelles.

III. Présentation du montage de test et du protocole de mesure

III.1. Préambule

Le premier enseignement que nous pouvons tirer de l'étude réalisée dans la précédente partie est que nous pouvons réduire notre domaine d'investigation aux systèmes à capacités commutées. L'idéal serait de réduire l'étude au simple échantillonneur situé en début de chaîne mais cela ne serait pas réaliste. Comme les montages résonateurs utilisés dans la conception des modulateurs Sigma Delta passe-bande temps discret sont de conception et d'étude complexes, nous nous sommes tournés vers l'utilisation d'un montage plus simple.

Le second enseignement que nous pouvons tirer de l'étude précédente est que le « siège » de génération du jitter interne se situe vraisemblablement dans le système de génération d'horloges non-recouvrantes pour la commande des interrupteurs. Ce système avait été décrit précédemment (cf. Chapitre I, IV.3). Celui qui est utilisé dans nos circuits a la particularité de comporter un diviseur de fréquence par deux en plus d'un buffer d'entrée et de la logique de génération des signaux de commande des interrupteurs CMOS :

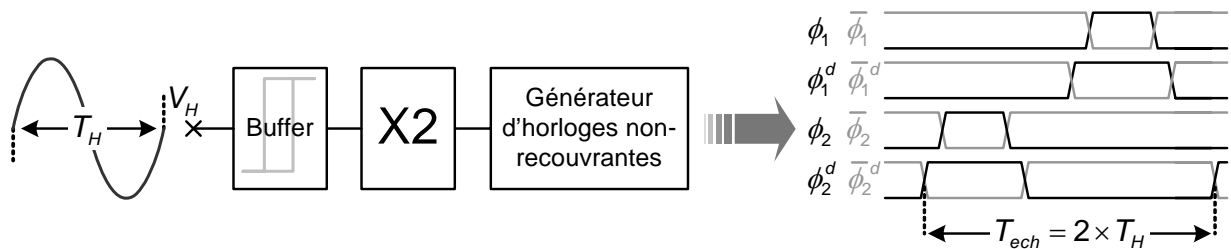


Figure 3 - 6 : Circuit générateur d'horloges non-recouvrantes

L'étude développée dans ce chapitre s'appuie sur une structure simple à capacités commutées réalisée à partir des mêmes éléments (Amplificateur, interrupteurs, capacités et surtout le générateur d'horloges non-recouvrantes) que ceux utilisés pour la conception des modulateurs. L'idée est de mesurer les caractéristiques du jitter interne au système et ensuite de les transposer à un système plus complexe tel un modulateur Sigma Delta passe-bande à temps discret.

III.2. La structure de test

Nous avons opté pour un circuit simple d'amplification comme structure de test. Ce montage décrit ci-dessous (cf. Figure 3 - 7), est un filtre à capacités commutées dont le gain est de 10 (gain linéaire) lorsque la fréquence d'entrée est égale à $\frac{1}{8}$ de la fréquence d'échantillonnage :

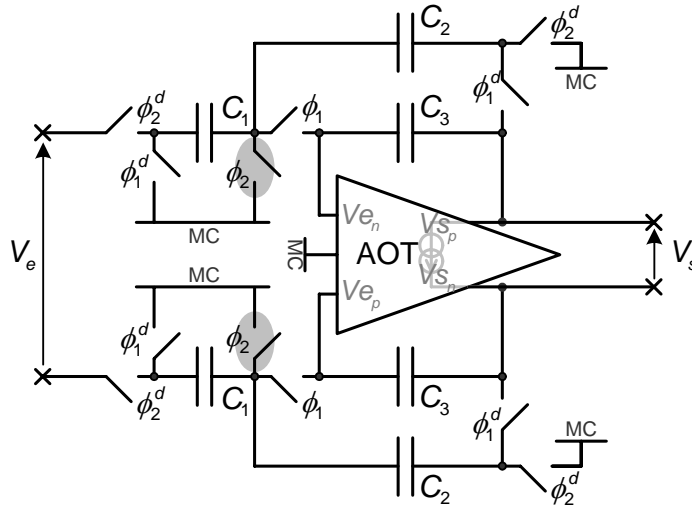


Figure 3 - 7 : Schéma du circuit « Gain_10 »

La fonction de transfert en z du montage est égale à :

$$H(z) = \frac{V_s}{V_e} = \frac{C_1 z^{-1}}{(C_2 + C_3) - C_3 z^{-1}} \quad \text{Eq. 3 - 2}$$

Ou encore :

$$V_s(n) = \alpha \cdot V_e(n-1) + \beta \cdot V_s(n-1) \quad \text{avec} \quad \alpha = \frac{C_1}{C_2 + C_3} \quad \text{et} \quad \beta = \frac{C_3}{C_2 + C_3} \quad \text{Eq. 3 - 3}$$

Les capacités C_1 , C_2 et C_3 ont été choisies pour obtenir le gain de 10 pour une fréquence de travail égale à $\frac{1}{8}$ de la fréquence d'échantillonnage (F_{ech}). Nous utilisons la transformation suivante :

$$z = e^{j \cdot \omega \cdot T_{ech}} \quad \text{Eq. 3 - 4}$$

La fonction de transfert devient alors :

$$\left| \frac{V_s}{V_e} \left(\frac{\omega_{ech}}{8} \right) \right|^2 = \left| \frac{C_1 \cdot e^{-j \cdot \frac{2 \cdot \pi}{8}}}{(C_2 + C_3) - C_3 \cdot e^{-j \cdot \frac{2 \cdot \pi}{8}}} \right|^2 \quad \text{Eq. 3 - 5}$$

Nous obtenons les performances souhaitées pour les valeurs de capacités $C_2 = C_3 = 1pF$ et $C_1 = 14.76pF$. Ce qui fait équivalent à $\alpha = 7.38$ et $\beta = 0.5$.

N.B. : même si cela ne transparaît pas, le choix de valeur de capacités est également conditionné par les caractéristiques de l'Amplificateur Opérationnel à Transconductance qui a été dimensionné pour charger et décharger une certaine gamme de valeurs de capacité. Nous rappelons qu'il est normalement utilisé pour d'autres types de circuit.

Ce circuit est conçu pour fonctionner avec une fréquence d'échantillonnage (deux fois inférieure à celle entrant dans le générateur d'horloges cf. Figure 3 - 6) comprise entre 100kHz et 13MHz. Dans ces conditions, les performances simulées (grâce au noyau de simulation SpectreRF) en gain et en bruit (en sortie et ramené en entrée) du montage « Gain_10 » pour une fréquence d'échantillonnage de 13MHz sont les suivantes :

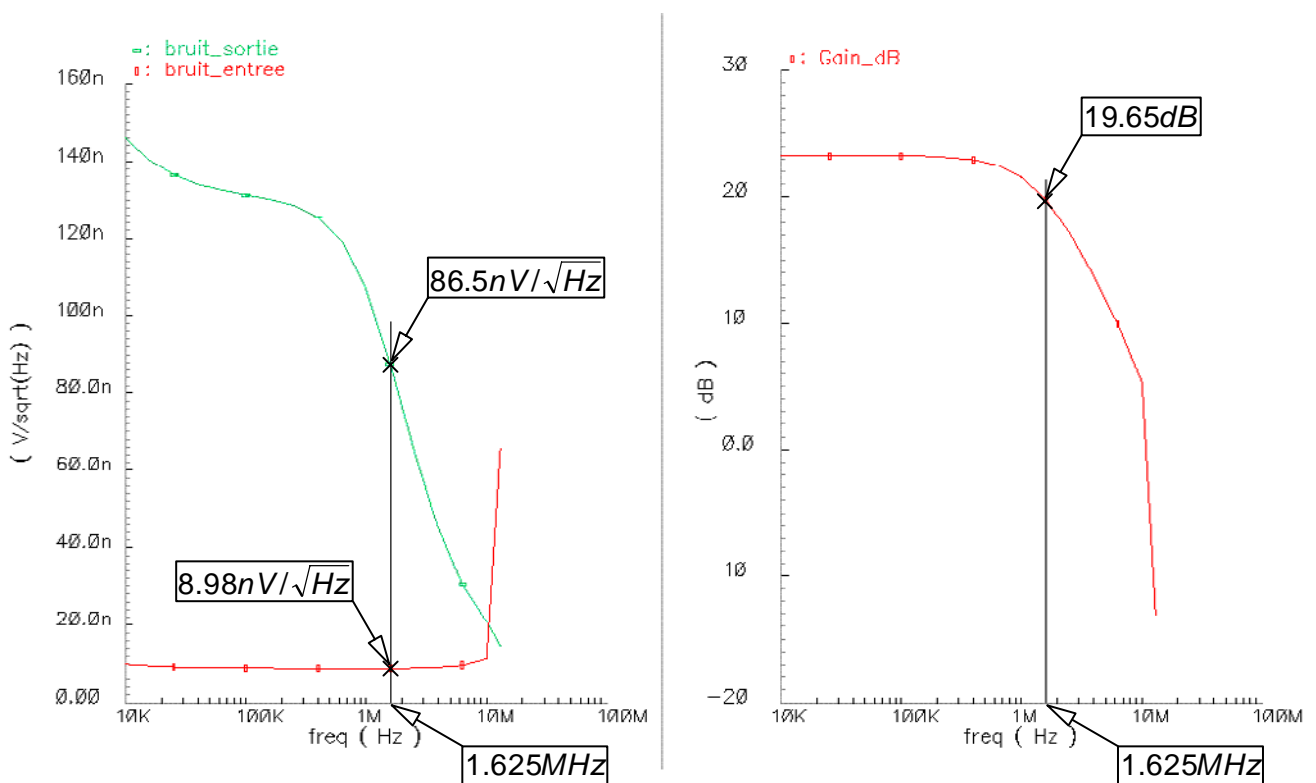


Figure 3 - 8 : Performances simulées du circuit « Gain_10 »

Le gain simulé à une fréquence d'entrée de 1.625MHz ($F_{ech}/8$) est égale à 19.65dB, le bruit ramené en entrée est de $8.98nV/\sqrt{Hz}$. Ce circuit a été implanté sur silicium en technologie AMS 0.35µm [AMS], la vue du layout est disponible en Annexe 1 (cf. §III.1). Ses performances mesurées à une fréquence d'entrée de 1.625MHz sont de 19.72dB pour le gain et de $8.98nV/\sqrt{Hz}$ pour le bruit ramené en entrée.

III.3. La méthode de mesure

Avant de nous lancer dans la description du protocole de mesure, nous devons effectuer deux mises au point nécessaires à la compréhension de la suite de cette partie :

Tout d'abord il est impossible de mesurer de petits décalages aléatoires des fronts d'horloge. Nous nous intéresserons donc uniquement aux effets du jitter à savoir aux variations de tension qu'il engendre lors de l'échantillonnage (cf. Chapitre II). Une fois ces variations de tension traitées, nous pourrons accéder au jitter.

Lors de l'étude des différents bruits effectuée dans le Chapitre II, nous avons utilisé une représentation fréquentielle des résultats. Cette représentation est couramment utilisée pour caractériser les performances des chaînes Radio Fréquence (cf. Chapitre I, §II). Naturellement nous aurions pu utiliser ce type de mesures pour la caractérisation du jitter de notre structure de test. Nous avons cependant opté pour une méthode d'analyse statistique et par là même nous situerons notre étude dans le domaine temporel.

III.3.1. Bilan des bruits

Comme nous venons de le dire dans cette introduction nous allons nous intéresser aux variations de tensions en sortie de notre banc de test pour remonter aux caractéristiques du jitter. Considérons la figure ci-dessous qui illustre un bilan (non exhaustif) des principales sources de bruit. Nous constatons que le bruit collecté en sortie du montage de test dépend non seulement du jitter interne, mais aussi du bruit des générateurs de signaux sinusoïdaux (sources externes) et également des bruits tels que le bruit thermique ou le bruit de grenaille (sources internes) :

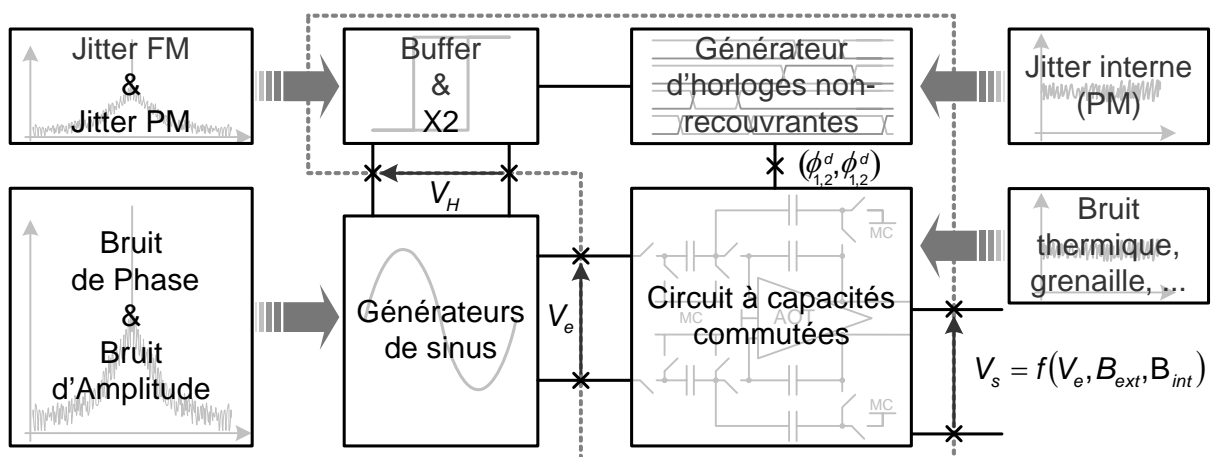


Figure 3 - 9 : Bilan des source de bruits

Le générateur d'horloges non-recouvrantes étant un générateur piloté par une horloge extérieure nous considérons que le jitter interne est de type PM. Conformément à l'étude théorique réalisée dans le Chapitre II, les sources internes de bruit (jitter PM, thermique, grenaille, ...) sont la combinaison de bruit de type blanc ou/et Flicker ($1/f$). Par soucis de simplification, nous considérons que toutes les sources internes sont des bruits blancs. A l'opposé les sources de bruit externes (générateur de sinus et de carrés) sont la combinaison de bruit de type blanc, de bruit de Flicker de bruit en $1/f^2$ et $1/f^3$.

Considérons le fonctionnement de notre filtre. Pour faciliter la compréhension nous avons concentré l'ensemble des perturbations liées au jitter en amont du filtre :

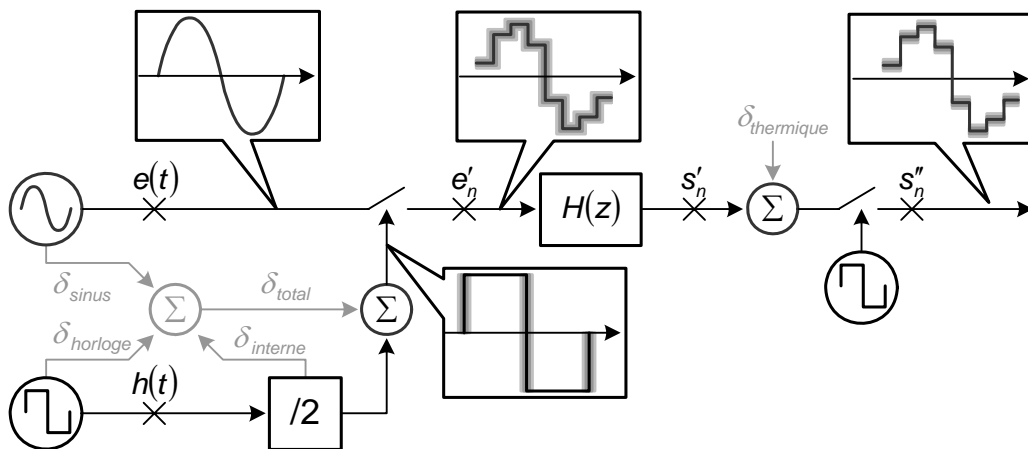


Figure 3 - 10 : Contribution des sources de bruit dans le filtre à capacités commutées

N.B. : Pour ne conserver que la variation d'amplitude, et donc éliminer la variation temporelle, nous effectuons un second échantillonnage de la sortie du filtre. La sortie du filtre étant un signal à temps discret il nous suffit pour cela d'échantillonner une fois que le signal est établi. Ce second échantillonnage n'aura alors aucune influence sur l'étude théorique qui va suivre.

Nous rappelons (cf. Chapitre II) que, du point de vue de la sortie d'un échantillonneur, l'échantillonnage idéal d'un signal sinusoïdal bruité peut se ramener à l'échantillonnage jitté d'un signal sinusoïdal idéal. Nous pouvons donc regrouper sous la dénomination δ_{total} la somme de l'ensemble des contributions en jitter des générateurs de signaux sinusoïdaux (δ_{sinus}), et carrés ($\delta_{horloge}$) ainsi que le jitter interne ($\delta_{interne}$) :

$$\delta_{total}^2 = \delta_{sinus}^2 + \delta_{horloge}^2 + \delta_{interne}^2 \tag{Eq. 3 - 6}$$

Comme le filtre que nous utilisons est un filtre récursif, la sortie à un instant donné s'_n dépend des états précédents du système $s'_{n-1, n-2, \dots}$ et du signal d'entrée $e'_{n-1, n-2, \dots}$:

$$\begin{aligned} s'_n &= \alpha \cdot e'_{n-1} + \beta \cdot s'_{n-1} \\ &= \alpha \cdot \sum_{i=0}^{n-1} \beta^i \cdot e'_{n-i-1} && \text{(cf. Eq. 3 - 1)} \\ &\approx \alpha \cdot \sum_{i=0}^{n-1} \beta^i \cdot e_{n-i-1} + \alpha \cdot \sum_{i=0}^{n-1} \beta^i \cdot \left. \frac{\partial e}{\partial t} \right|_{t_{n-i-1}} \cdot \delta_{total}^{n-i-1} \end{aligned} \quad \text{Eq. 3 - 7}$$

e_n , T_{ech} et $\left. \frac{\partial e}{\partial t} \right|_{t_n}$ désignent respectivement le signal issu d'un échantillonnage idéal de $e(t)$, la période d'échantillonnage et la dérivée du signal $e(t)$ à l'instant d'échantillonnage $t_n = n \cdot T_{ech}$. Nous rappelons que dans notre cas $T_{ech} = 2 \cdot T_H$ où T_H désigne la fréquence de l'horloge.

Le bruit interne au circuit (thermique, grenaille, ...) $\delta_{thermique}$ s'ajoute ensuite au signal issu du filtre :

$$(s_n^n)^2 = (s'_n)^2 + (\delta_{thermique}^n)^2 \quad \text{Eq. 3 - 8}$$

Cette dernière contribution est facile à isoler. En effet en l'absence de signal d'entrée ($e(t) = 0$) la sortie est alors égale au bruit thermique, il suffit alors de retrancher cette valeur pour isoler s'_n .

Il n'en va pas de même pour déterminer la valeur de δ_{total} . En effet bien que la période d'échantillonnage T_{ech} et les paramètres du filtre (α, β) soient connus, nous ne maîtrisons pas la valeur de la pente du signal ce qui est très handicapant. De plus, en considérant l'équation précédente (cf. Eq. 3 - 7), nous constatons qu'il est impossible d'extraire la valeur instantanée à l'instant t_n du jitter total δ_{total}^n du fait de la récursivité de la suite. Et quand bien même nous connaîtrions cette valeur instantanée, il semble impossible à première vue d'isoler la contribution du jitter interne ($\delta_{interne}$) au sein du jitter total (δ_{total}) sans une analyse poussée des caractéristiques en bruit des deux générateurs. Etude que nous ne sommes pas en mesure d'effectuer.

Ainsi pour remonter aux caractéristiques du jitter nous allons devoir faire le tri des différents bruits pour isoler le bruit induit par le jitter interne. La méthode que nous proposons permet, de manière astucieuse, de faire cette sélection.

III.3.2. L'histogramme verrouillé

Cette méthode présentée dans [DMB94] et basée sur les travaux de [SAW90] était initialement destinée à la caractérisation du jitter interne des CAN. Moyennant quelques aménagements, nous allons l'employer pour caractériser le jitter interne du circuit de test.

Le principe de ce protocole de mesure consiste à appliquer la même fréquence de travail pour le signal d'entrée et pour le signal d'échantillonnage (soit une fréquence d'horloge deux fois supérieure). Nous réalisons ensuite une étude statistique de la sortie du montage de test (s''_n) afin d'obtenir une caractérisation du bruit. Par souci de simplification, la loi de probabilité ($p_{s''}$) de s''_n sera assimilée à une gaussienne $N(m_{s''}, \sigma_{s''})$ dont la valeur moyenne ($m_{s''}$) et surtout de l'écart type ($\sigma_{s''}$) nous permettra de remonter aux caractéristiques du jitter interne.

Considérons la figure ci-dessous. L'échantillonnage d'une sinusoïde ayant la même fréquence que horloge d'échantillonnage produit, lorsque l'horloge est jittée, un signal e'_n qui « oscille » autour d'un signal continu e_n qui correspond à un échantillonnage idéal :

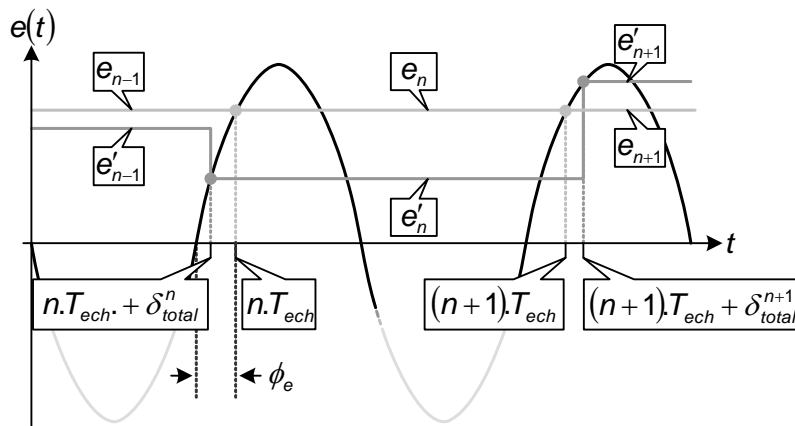


Figure 3 - 11 : Echantillonnage jitté d'une sinusoïde

L'expression mathématique des signaux e_n et e'_n est la suivante :

$$\begin{aligned}
 e_n = A \cdot \sin(2 \cdot \pi \cdot F_{ech} \cdot (n \cdot T_{ech}) + \phi_e) & \quad \left| \quad e'_n = A \cdot \sin(2 \cdot \pi \cdot F_{ech} \cdot (n \cdot T_{ech} + \delta_{total}^n) + \phi_e) \right. \\
 = A \cdot \sin(\phi_e) & \quad \left. \approx e_n + A \cdot 2 \cdot \pi \cdot F_{ech} \cdot \cos(\phi_e) \cdot \delta_{total}^n \right.
 \end{aligned}
 \tag{Eq. 3 - 9}$$

ϕ_e désigne le déphasage entre le signal d'échantillonnage et la sinusoïde.

La pente peut maintenant être considérée comme constante. Nous pouvons dès lors réécrire l'Eq. 3 - 7 sous une forme plus exploitable :

$$s'_n \approx \alpha \cdot \sum_{i=0}^{i=n-1} \beta^i \cdot e_{n-i-1} + \alpha \cdot A \cdot 2 \cdot \pi \cdot F_{ech} \cdot \cos(\phi_e) \sum_{i=0}^{i=n-1} \beta^i \cdot \delta_{total}^{n-i-1} \quad \text{Eq. 3 - 10}$$

Ainsi nous obtenons en sortie du filtre un signal s'_n qui « oscille » autour d'une position d'équilibre qui correspondrait à un échantillonnage idéal. L'écart entre la valeur idéale et la valeur mesurée correspond alors à la contribution du jitter total.

Grâce au déphase ϕ_e il est maintenant possible de « choisir » la pente du signal. Afin d'avoir une valeur plus facilement mesurable (la plus grande) en sortie du filtre nous prendrons la pente maximum c'est-à-dire celle obtenue par un échantillonnage autour du passage à zéro de la sinusoïde d'entrée ($\phi_e = 0$).

Intéressons nous maintenant aux propriétés statistiques des diverses sources de bruit :

- Le jitter interne (jitter PM) et le bruit thermique seront modélisés par une distribution statistique gaussienne respectivement $N(0, \sigma_{interne})$ et $N(0, \sigma_{thermique})$.
- Les contributions au jitter total des générateurs de sinusoïdes et de signaux carrés seront représentées par des pseudo gaussiennes dont l'écart type est dépendant de la fréquence des générateurs (cf. Eq. 3 - 15) et d'autres paramètres qui n'entrerons pas en compte dans notre étude et que nous désignerons par la lettre γ : $N(0, \sigma_{sinus}(F_{ech}, \gamma))$ et $N(0, \sigma_{horloge}(F_{ech}, \gamma))$.

La loi de probabilité du jitter total est alors une gaussienne de moyenne nulle et d'écart type égal à :

$$\sigma_{total}^2(F_{ech}) = \sigma_{sinus}^2(F_{ech}, \gamma) + \sigma_{horloge}^2(F_{ech}, \gamma) + \sigma_{interne}^2 \quad \text{Eq. 3 - 11}$$

Comme le bruit thermique, qui s'ajoute à la sortie du filtre, a une valeur moyenne nulle, nous pouvons d'ors et déjà donner l'expression de la valeur moyenne ($m_{s''}$) de la gaussienne qui régit la loi de probabilité de la sortie du montage de test s''_n ainsi (cf. Eq. 3 - 9 et Eq. 3 - 10) :

$$m_{s''} = m_{s'} = \alpha \cdot A \cdot \sin(\phi_e) \sum_{i=0}^{i=n-1} \beta^i \quad \text{Eq. 3 - 12}$$

N.B. : la pente maximum sera obtenue pour une sortie de valeur moyenne nulle.

Concentrons nous maintenant sur l'écart type ($\sigma_{s''}$). L'Eq. 3 - 8 nous permet d'extraire la contribution des bruits internes (autres que le jitter interne) :

$$\sigma_{s''}^2 = \sigma_{s'}^2 + \sigma_{thermique}^2 \quad \text{Eq. 3 - 13}$$

Où $\sigma_{s'}$ désigne l'écart type de la gaussienne qui régit la loi de probabilité de la sortie du filtre s'_n .

Toute la difficulté va maintenant résider dans l'établissement d'une loi liant les écarts types $\sigma_{s'}$ et σ_{total} . Nous avons opté pour une approche combinant théorie et modélisation. Cette méthode est décrite dans l'Annexe 2 (cf. III.2).

Nous obtenons alors l'approximation suivante :

$$\sigma_{s'}^2 = (A.2.\pi.F_{ech}.G)^2 . \sigma_{total}^2 \quad \text{Eq. 3 - 14}$$

Où G est une constante définie dans l'Annexe 2.

Maintenant que nous disposons d'une approximation nous permettant d'isoler le jitter total ($\delta_{total}(F_{ech})$) il nous ne reste plus qu'à extraire la contribution du jitter interne ($\delta_{interne}$) de cette expression. Pour cela nous utilisons une méthode de résolution graphique.

D'après BARNES et al. [BCC71] le jitter des générateurs décroît proportionnellement à la fréquence du générateur ce qui veut dire pour nous :

$$\sigma_{sinus / horloge}^2 (F_{ech}, \gamma) = \frac{C}{F_{ech}^2} \quad \text{Eq. 3 - 15}$$

C est une constante définie par les caractéristiques de l'oscillateur.

De plus afin de garantir un déphasage constant entre l'entrée et l'horloge, les générateurs de signaux (sinusoïdaux et carrés) doivent être synchronisés. Leurs contributions en bruit seront alors corrélées puisque le rapport entre ces deux générateurs sera celui de maître/esclave.

Ainsi en effectuant une série de mesures à diverses fréquences de travail nous devrions obtenir une courbe semblable à celle présentée ci-dessous :

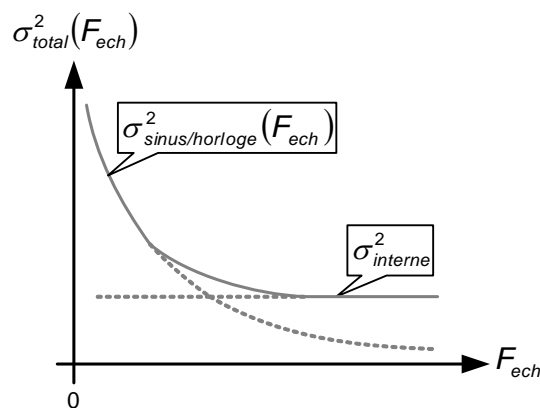


Figure 3 - 12 : Ecart type σ_{total} du jitter total fonction de la fréquence de travail

Nous pourrions alors isoler graphiquement la valeur de l'écart type $\sigma_{interne}$ du jitter interne.

Nous venons d'exposer dans cette partie une méthode de mesure des caractéristiques statistiques du jitter interne qui ne nécessite pas une connaissance des performances en bruit des générateurs. Bien qu'en théorie cette méthode fonctionne quelles que soient les performances des générateurs, nous avons un circuit de test dont la plage de fréquence d'utilisation (de quelques centaines de kHz à quelques MHz) n'est pas infinie. Nous prendrons donc soin, par la suite, de choisir des générateurs performants pour obtenir une courbe semblable à celle présentée ci-dessus.

Nous illustrons, ci-dessous, les différentes étapes de l'extraction de la contribution du jitter interne à partir de la mesure du bruit en sortie du montage de test :

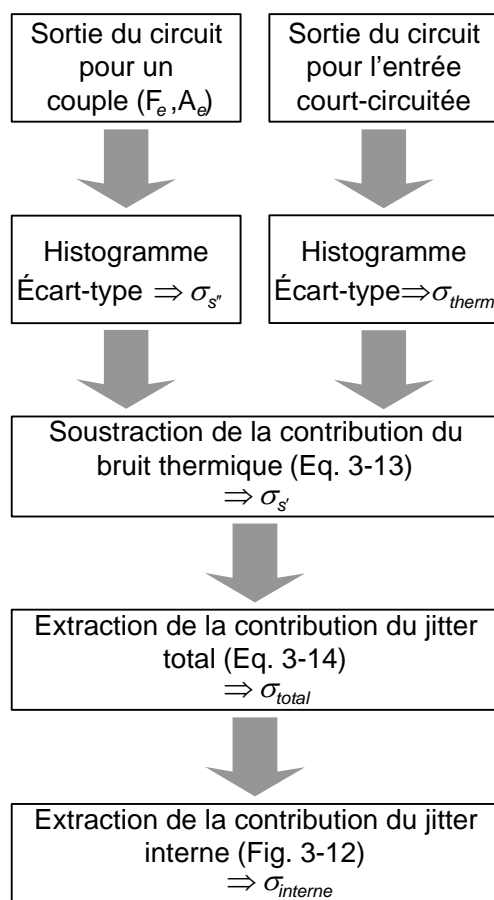


Figure 3 - 13 : Récapitulatif de la méthode de mesure

Passons maintenant de l'étude théorique à la mise en pratique.

IV. Mesures et interprétations

IV.1.1. Présentation du banc de mesure

Le schéma ci-dessous représente les principaux appareils auxquels nous avons fait appel :

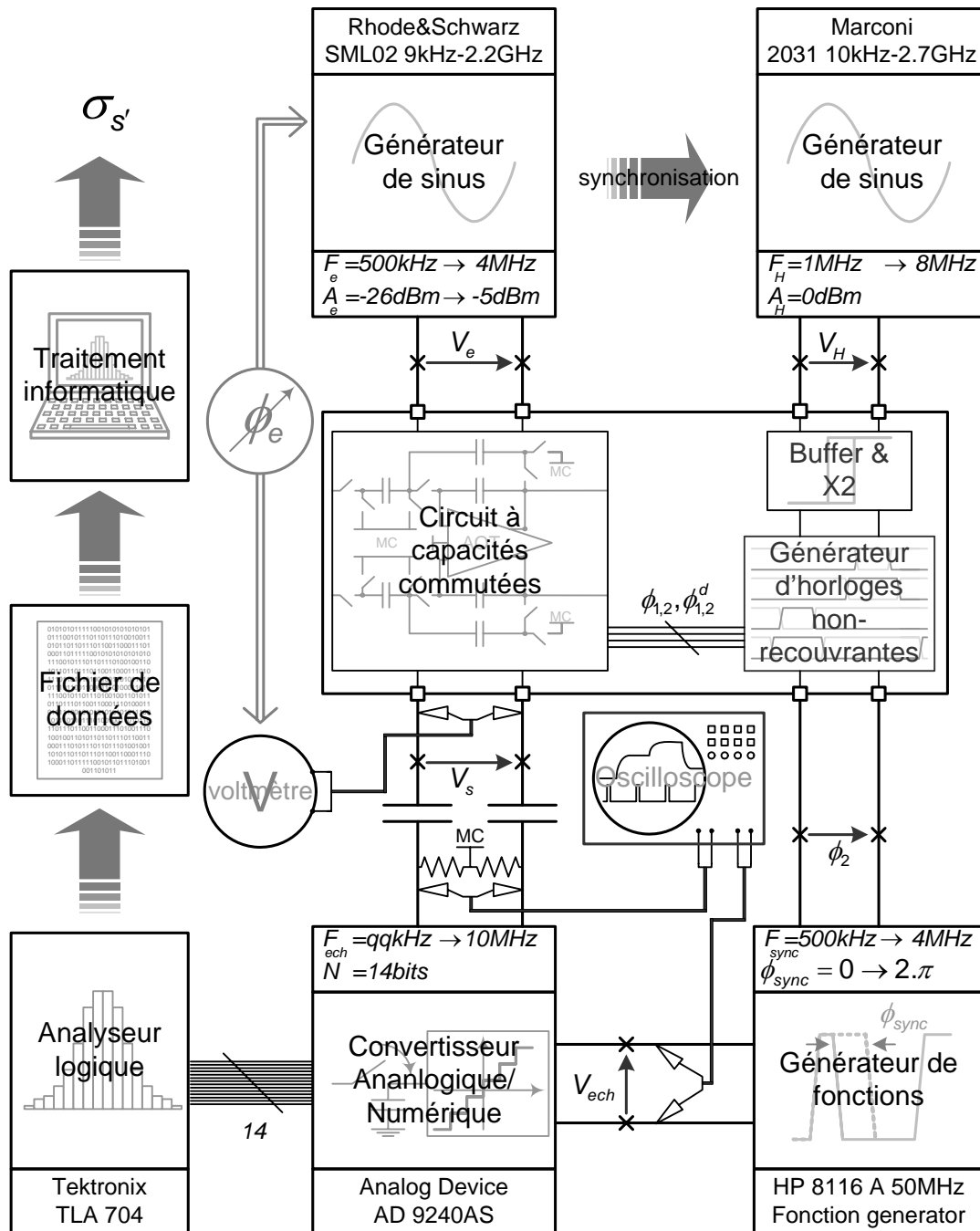


Figure 3 - 14 : Banc de test

Nous utilisons un premier générateur de signaux sinusoïdaux (Rhode&Schwarz SML02) pour le signal d'entrée. Nous ferons varier sa fréquence F_e de 500kHz à 4MHz par pas de 500kHz. Nous ferons également varier son amplitude A_e de $-26dBm$ à $-5dBm$ par pas de $-3dBm$ pour augmenter le nombre de mesures et par là même affiner notre résultat. Le second générateur (Marconi 2031) est utilisé pour le signal d'horloge. Il délivre également des signaux sinusoïdaux qui sont transformés par le buffer en signaux carrés utilisables par le générateur d'horloges non-recouvrantes. Les deux générateurs externes sont synchronisés pour maintenir un déphasage constant entre les deux signaux d'entrées. Nous réglons le déphasage ϕ_e en mesurant la valeur moyenne de la sortie du filtre à l'aide d'un voltmètre (cf. Eq. 3 - 12). Nous surveillerons en permanence cette mesure afin d'ajuster le déphasage d'une mesure sur l'autre. Nous rappelons que nos approximations sont valables pour une pente maximum soit un déphasage entre le signal d'entrée et l'horloge d'échantillonnage correspondant à une sortie moyenne nulle.

Nous devons numériser les valeurs de la sortie du filtre à capacités commutées afin de les traiter numériquement pour en extraire l'écart type du jitter interne. Nous utilisons pour cela un Convertisseur Analogique/Numérique de marque Analog Device l'AD9240AS [AD9240]. Ce convertisseur a une plage de fonctionnement en tension comprise entre $\pm 1V$ et fournit une sortie codée sur 14bits. De plus sa fréquence de travail est comprise entre quelques centaines de kHz et quelques MHz, ce qui est notre plage de fonctionnement. Ce convertisseur est monté sur une carte d'évaluation fournie par le fabricant.

Lors de la conception de notre circuit nous avons prévu une sortie d'une des horloges internes (en l'occurrence la phase ϕ_2) pour d'éventuelles mesures à l'analyseur de spectre. Ces mesures n'ont pas été réalisées au regard de la faible sortance du générateur interne (incapable de « driver » une charge de 50Ω). Néanmoins cette sortie va quand même être utilisée pour synchroniser le CAN. En effet pour numériser correctement la sortie du filtre nous devons effectuer l'échantillonnage (interne au CAN) une fois que la sortie est établie. Nous devons donc nous synchroniser avec l'horloge interne de la puce. Comme le signal ϕ_2 est trop faible nous utilisons un générateur de fonction (HP 8116) pour amplifier ce signal afin de nous synchronisons sur cette phase. En jouant sur le déphasage de cet appareil nous pouvons « choisir » notre moment d'échantillonnage. Nous visualisons la synchronisation grâce à un oscilloscope.

La sortie du CAN est connectée à un analyseur logique (Tektronix TLA704) qui a une capacité de stockage de $2^{15} \approx 32k$ échantillons. Chaque acquisition est stockée sous forme de fichier texte qui est exploité sous le logiciel Excel. De là nous extrayons l'écart type du jitter interne en utilisant les

équations développées dans la partie précédente. Afin d'augmenter la fiabilité de nos mesures nous réalisons 20 acquisitions par couple $(F_e; A_e)$.

IV.1.2. Mesure type

Nous prendrons l'exemple du couple $(F_e = 2MHz, A_e = -14dBm)$. Une fois les variables d'amplitude et de fréquence correctement configurées, la première mesure consiste à déterminer le bruit thermique. Pour cela nous court-circuitons l'entrée avec un bouchon d'impédance 50Ω . La figure ci-dessous illustre les histogrammes obtenus à partir des 20 acquisitions :

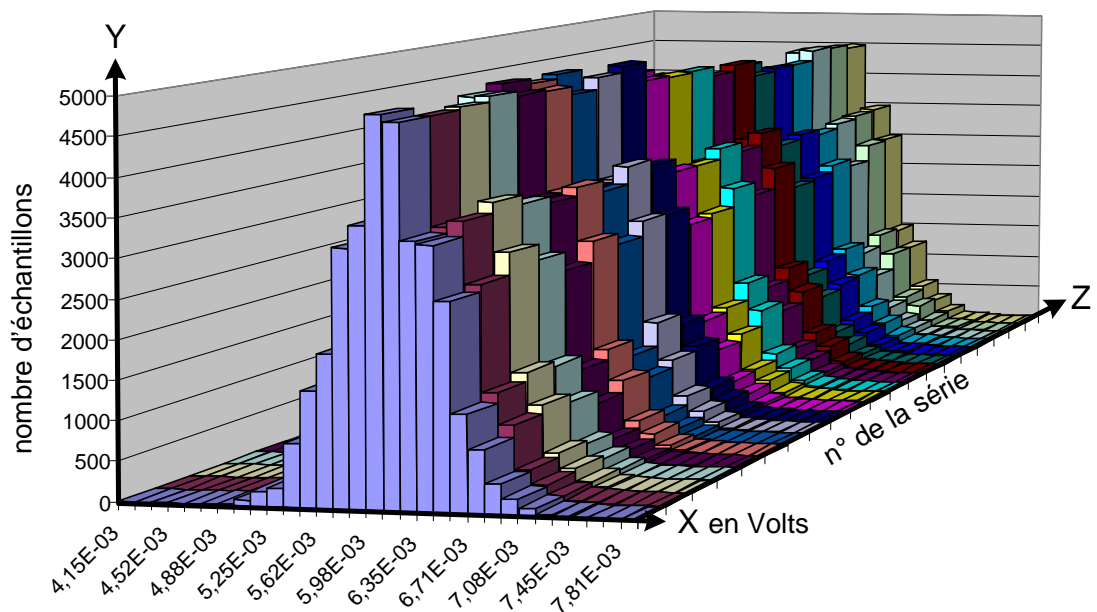


Figure 3 - 15 : Histogrammes du bruit thermique

L'axe des x représente les tensions en V obtenues à partir des valeurs codées multipliées par le pas du CAN ($q = 2V/2^{14}$). L'axe de y représente la répartition des 32768 échantillons et l'axe des z représente les 20 acquisitions.

Les écarts types de chacune des 20 acquisitions sont rassemblés dans le tableau suivant (les valeurs sont données en μV) :

n°	1	2	3	4	5	6	7	8	9	10
σ	360,3	360,9	361,7	353,0	351,8	356,7	354,0	356,8	362,3	365,4
n°	11	12	13	14	15	16	17	18	19	20
σ	370,8	374,8	379,4	376,5	380,3	382,2	389,0	377,2	382,1	379,2

Tab 3 - 1

Nous prenons la moyenne arithmétique et trouvons alors :

$$\sigma_{thermique} = 368,8 \mu V \quad \text{Eq. 3 - 16}$$

N.B. : nous effectuerons cette mesure à chaque changement de fréquence

Une fois la valeur de l'écart type du bruit thermique obtenue, nous branchons le générateur de tension d'entrée. Nous vérifions que le déphasage entre l'horloge et le signal est correct au voltmètre, puis nous réglons le signal de synchronisation du CAN. Nous pouvons alors réaliser pour chaque amplitude une série de 20 acquisitions.

Pour le couple ($F_e = 2MHz, A_e = -14dBm$) les histogrammes obtenus à partir des 20 acquisitions sont représentés ci-dessous :

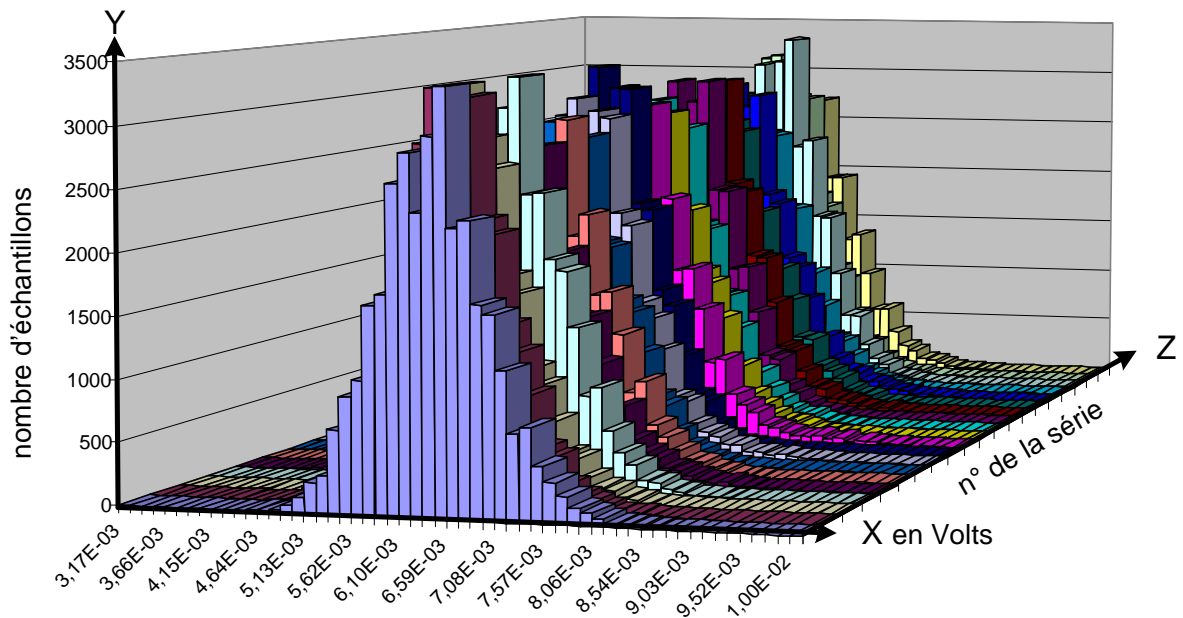


Figure 3 - 16 : Histogrammes du bruit à la sortie du filtre

Les écarts types de chacune des 20 acquisitions sont rassemblés dans le tableau suivant (les valeurs sont données en μV) :

n°	1	2	3	4	5	6	7	8	9	10
σ	566,9	524,4	539,6	555,8	642,4	640,1	582,6	592,3	520,6	665,1
n°	11	12	13	14	15	16	17	18	19	20
σ	615,2	605,0	557,8	586,5	640,5	618,5	609,8	527,9	549,4	625,7

Tab 3 - 2

Nous prenons la moyenne arithmétique :

$$\sigma_{sr} = 588,3 \mu V \quad \text{Eq. 3 - 17}$$

Pour remonter à la valeur de l'écart type du jitter interne nous utilisons alors les équations définies dans la partie précédente (cf. Eq. 3 - 13 et Eq. 3 - 14) :

$$\sigma_{s'} = \sqrt{\sigma_{s''}^2 - \sigma_{thermique}^2} = 458,3 \mu V$$

$$\sigma_{total} = \frac{\sigma_{s'}}{2 \times \pi \times 8.2 \times 2e^6 \times \sqrt{2 \times 50 \times 1e^{-3} \times 10^{-14}}} = 70,5 ps \quad \text{Eq. 3 - 18}$$

IV.1.3. Généralisation des mesures

Le reste des mesures s'effectue de manière identique, nous obtenons alors la courbe suivante :

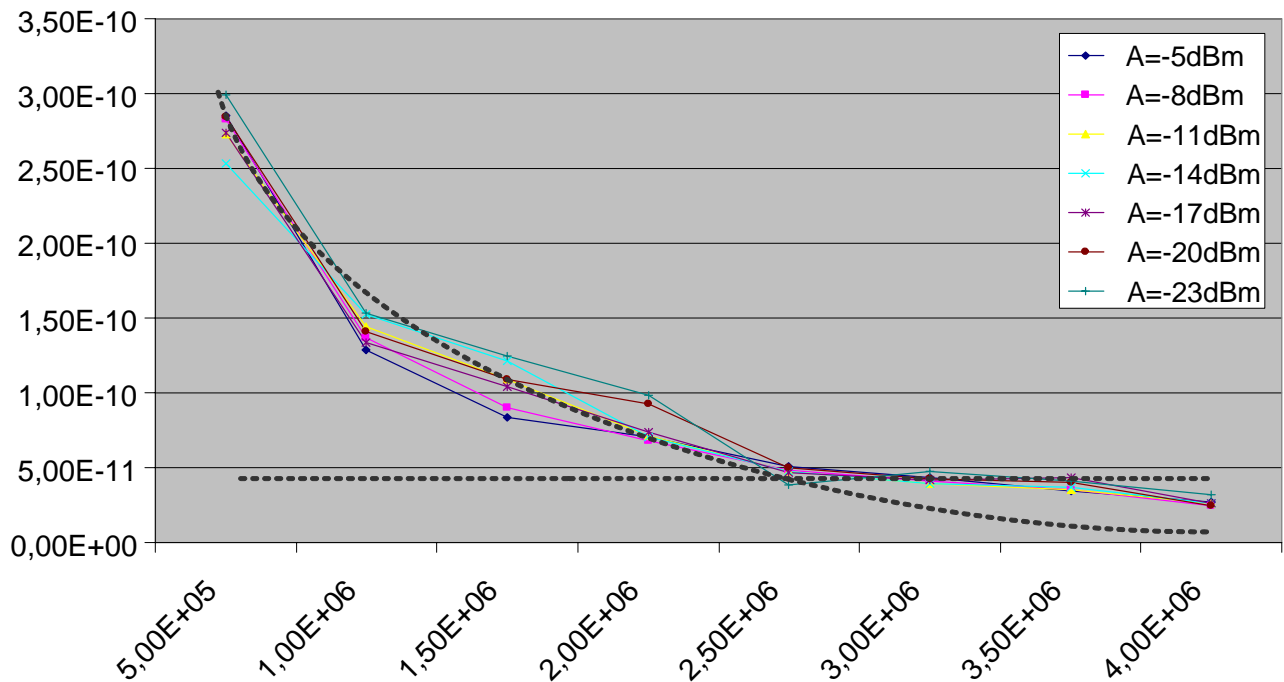


Figure 3 - 17 : Ecart type σ_{total} du jitter total fonction de la fréquence de travail

Nous avons retiré du graphique la courbe obtenue pour l'amplitude $A_e = -25dBm$ car le bruit de jitter est inférieur au bruit thermique et n'est donc pas mesurable.

Nous constatons néanmoins que la courbe ci-dessus présente un aspect général semblable à la courbe théorique Figure 3 - 12. Nous pouvons donc estimer graphiquement la valeur de l'écart type du jitter interne, nous trouvons alors :

$$35 ps \leq \sigma_{interne} \leq 45 ps \quad \text{Eq. 3 - 19}$$

V. CONCLUSION

Le but de ce chapitre était d'évaluer ou d'approximer la valeur du jitter interne d'une structure simple utilisant le principe des capacités commutées. Après une rapide étude des effets du jitter dans un modulateur Sigma Delta nous avons pu localiser le bloc le plus sensible aux variations aléatoires des fronts d'horloge : la structure à capacités commutées. Nous avons donc réalisé une structure de test utilisant les mêmes briques (notamment le générateur d'horloges non-recouvrantes) que celle utilisées dans la conception des modulateurs. Une étude un peu plus poussée de notre montage nous a ensuite révélé que outre le jitter interne, d'autres sources internes ou externes de bruit entraient dans la composition du bruit en sortie de filtre. Nous avons alors exposé une méthode d'étude statistique qui permettait d'isoler le jitter interne parmi toutes les autres sources de bruit. La dernière partie a été consacrée à la mise en œuvre de cette méthode. A l'issue des mesures nous avons pu fixer une plage de valeur pour le jitter interne. Nous pouvons donc maintenant réutiliser cette valeur dans une structure plus complexe et extrapoler ainsi le comportement d'un modulateur Sigma Delta vis-à-vis des divers bruits. Ce sera le but du chapitre suivant.

CHAPITRE IV :

Modélisation fonctionnelle en VHDL-AMS
d'une chaîne de réception Radio Fréquence

I. INTRODUCTION

Ce chapitre concerne l'étude des performances et la modélisation de la chaîne de réception radio-fréquence développée par la société EADS. Cette chaîne de réception est intégrée au sein d'une puce, représentée par la Figure 4 - 1 sous forme de blocs fonctionnels. Nous nous intéressons à la partie analogique qui comprend un Amplificateur à Contrôle Automatique de Gain (CAG), deux Mélangeurs commandés par un générateur de Quadrature lui-même commandé par la PLL, un Filtre Anti-Repliements, ainsi qu'un modulateur Sigma Delta.

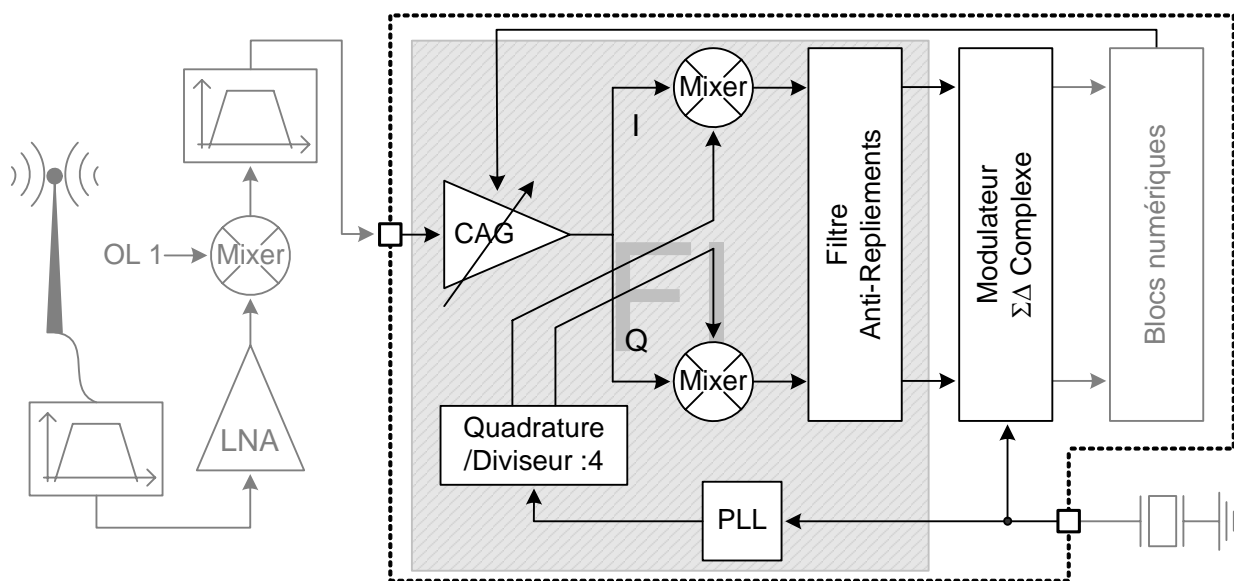


Figure 4 - 1 : Schéma Bloc de la puce RX contenant la chaîne de réception RF

Lors des précédentes réalisations sur silicium, les mesures ont laissé apparaître une divergence entre les résultats escomptés et ceux obtenus, principalement lorsqu'un signal de forte amplitude est appliqué en début de chaîne. Cette dégradation proportionnelle à l'amplitude du signal d'entrée, est caractéristique des bruits de type jitter. La finalité de ce chapitre est de mettre en évidence les contributions des sources diverses de bruit, en particulier l'influence du jitter d'horloge sur l'ensemble de la chaîne de réception.

Pour cela, la première partie du chapitre détaille, après une courte étude théorique, la modélisation en VHDL-AMS des principales caractéristiques (Gain, impédance d'entrée, bruit interne) des blocs qui constituent la chaîne FI (Fréquence Intermédiaire) : l'amplificateur à Contrôle Automatique de Gain, le Mélangeur, le générateur de quadrature et le filtre anti-repliements. Chaque modèle en VHDL-AMS sera comparé individuellement avec le modèle à base de transistors. Nous concluons

cette première partie par une analyse fréquentielle (FFT) de la chaîne FI afin de vérifier la bonne conformité ou non de nos modèles vis-à-vis des performances attendues.

La seconde partie concerne la modélisation des blocs qui constituent le modulateur Sigma Delta : le générateur d'horloges non-recouvrantes, le CAN, le CNA et le résonateur. Comme pour la première partie chaque modèle en VHDL-AMS aura ses principales caractéristiques comparées avec le modèle transistor. Cette partie se conclue également par une simulation globale du modulateur.

La troisième partie, présente les résultats de simulation de la chaîne analogique complète (associant la chaîne FI et le modulateur Sigma Delta) et les compare avec les mesures.

II. Préambule

La première justification de la réalisation de nos modèles comportementaux en VHDL-AMS est purement industrielle. Ces modèles entrent directement dans le flot de conception ASIC et sont utilisés lors de simulations globales lors de l'élaboration d'un ou plusieurs blocs, en remplacement du modèle transistor. Ils doivent donc respecter le même brochage, être plus rapides en simulation que les modèles transistor et surtout être interactifs en indiquant à l'utilisateur, par exemple, si la connectique est correcte.

En outre, la réalisation de ces modèles est une étape clé de ma thèse. Ces modèles sont utilisés pour réaliser une simulation de l'ensemble de la chaîne de réception et ainsi mettre en évidence le rôle du jitter dans cette structure. Cette simulation complète est impossible en utilisant une description au niveau transistor. Les modèles comportementaux, dans ce cas, devront reprendre les principales caractéristiques des blocs et abandonner tout ce qui est superflu (calcul du courant consommé, vérification de la bonne alimentation, impédance d'entrée/sortie pour certain bloc, etc...) afin d'accroître la vitesse de simulation.

Nous avons donc développé des modèles polyvalents. Afin de garder la possibilité d'augmenter la précision et pour assurer la compatibilité broche à broche avec les montages au niveau transistor, les modèles sont composés de plusieurs couches hiérarchisées comme des poupées russes. Chaque modèle fait alors appel à un sous-modèle de complexité moindre, comme illustré ci-dessous :

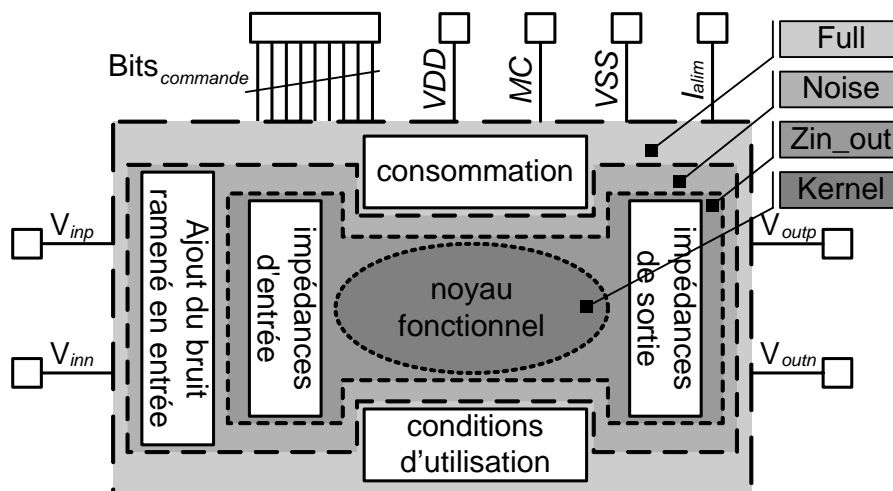


Figure 4 - 2 : Différents niveaux de modélisation

Le noyau fonctionnel (Kernel) regroupe l'ensemble des fonctionnalités « vitales » du modèle. Nous utiliserons ce modèle pour nos simulations globales de la chaîne de réception. Ce modèle est appelé par le modèle supérieur (Zin_out) où sont prises en compte les impédances d'entrée / sortie ce

dernier est lui même utilisé par le modèle incluant le bruit (Noise), lui-même étant appelé par le modèle complet (Full) qui implémente la consommation et la vérification des conditions d'utilisation. Un exemple de modélisation est donné ci-dessous :

```

-----
ENTITY MODELE IS
  GENERIC(      noise      : REAL := 0.8e-9;
              Tsmpt       : REAL := 1.0e-6);
  PORT (  TERMINAL VDD, VSS, MC : ELECTRICAL;
         TERMINAL Ialim       : ELECTRICAL;
         TERMINAL INP, INN    : ELECTRICAL;
         TERMINAL OUTP, OUTN  : ELECTRICAL;
         SIGNAL  Commande     : IN BIT_VECTOR (2 DOWNTO 0));
END ENTITY MODELE;

```

DEFINITION DES paramètres et des connexions : alimentation en tension et courant - entrée différentielle - sortie différentielle - bit de commande

----- FULL -----

```

ARCHITECTURE behaviour_full OF MODELE IS
  QUANTITY V_alim  ACROSS  I_alim  THROUGH  VDD  TO VSS;
  QUANTITY V_mc    ACROSS  I_bias  THROUGH  MC   TO VSS;
  QUANTITY V_bias  ACROSS  I_bias  THROUGH  Ialim TO VSS;

```

DEFINITION DU COUPLE TENSION / COURANT POUR LES NŒUDS INTERNES ET EXTERNES

```

SIGNAL time : BIT := '0';

```

```

BEGIN

```

```

PROCESS (  V_alim'ABOVE(2.9),V_alim'ABOVE(3.6),I_bias'ABOVE(80.0e-6),
          I_bias'ABOVE(120.0e-6),V_mc'ABOVE(v_alim*0.525),
          V_mc'ABOVE(v_alim*0.475),time)

```

VERIFICATION DES ALIMENTATIONS. Nous utilisons un signal (time) décalé d'une 1fs pour que la vérification des tensions s'effectue après l'analyse DC.

```

BEGIN
IF NOW > 0.0 THEN
ASSERT (v_alim > 2.9)      REPORT "VOLTAGE SUPPLY UNDER 2.9V"
SEVERITY WARNING;
ASSERT (v_alim < 3.6)    REPORT "VOLTAGE SUPPLY OVER 3.6V"
SEVERITY WARNING;
ASSERT (I_bias > 80.0e-6) REPORT "CURRENT SUPPLY TOO LOW"
SEVERITY WARNING;
ASSERT (I_bias < 120.0e-6) REPORT "CURRENT SUPPLY TOO HIGH"
SEVERITY WARNING;
ASSERT (V_mc < v_alim*0.525) REPORT "COMMON VOLTAGE SUPPLY TOO HIGH"
SEVERITY WARNING;
ASSERT (V_mc > v_alim*0.475) REPORT "COMMON VOLTAGE SUPPLY TOO LOW"
SEVERITY WARNING;
ELSE time <= '1' AFTER 1fs; END IF;
END PROCESS;

```

```

I_bias ==V_bias/10.0e+3;

```

CALCUL DU COURANT CONSOMME

```

M : ENTITY MODELE (behaviour_noise)
  GENERIC MAP ( noise =>noise, Tsmpt=>Tsmpt)
  PORT MAP (  VDD=>VDD, VSS=>VSS, MC=>MC, Ialim=>Ialim,
             INP=>INP, INN=>INN, OUTP=>OUTP, OUTN=>OUTN,
             Commande => Commande);

```

```

END ARCHITECTURE behaviour_full;

```

----- NOISE -----

```

ARCHITECTURE behaviour_noise OF MODELE IS

```

```

TERMINAL INP_noise, INN_noise : ELECTRICAL;
QUANTITY Vp_noise  ACROSS  Ip_noise THROUGH  INP  TO INP_noise;
QUANTITY Vn_noise  ACROSS  In_noise THROUGH  INN  TO INN_noise;
SIGNAL  White_n : REAL := 0.0;

```

AJOUT D'UN BRUIT BLANC : cette couche

```

BEGIN
  noise_addition : PROCESS
  VARIABLE white : REAL_VECTOR (0 TO 4) := (11.0, 4862.0, 214.0, 546.0, 1105.0));
  BEGIN
    white := white_noise_Vrms(1.0/Tsmp, noise, white);
    White_n <= white(0);
  WAIT FOR (Tsmp);
  END PROCESS noise_addition;

  Vp_noise == +White_n'RAMP(Tsmp/10.0, Tsmp/10.0)/2.0;
  Vn_noise == -White_n'RAMP(Tsmp/10.0, Tsmp/10.0) /2.0;

  M : ENTITY MODELE (behaviour_Zin_out)
  GENERIC MAP ( noise =>noise_Hz, Tsmp=>Tsmp)
  PORT MAP (   VDD=>VDD, VSS=>VSS, MC=>MC, lalim=>lalim,
              INP=>INP_noise, INN=>INN_noise,
              OUTP=>OUTP, OUTN=>OUTN, Commande => Commande);

END ARCHITECTURE behaviour_noise;

----- Zin/Zout -----

ARCHITECTURE behaviour_Zin_out OF MODELE IS

CONSTANT Cin      : REAL := 0.6e-12;
CONSTANT Cout     : REAL := 0.8e-12;
CONSTANT Rout     : REAL := 900.0;

QUANTITY Vin  ACROSS lin          THROUGH INP      TO INN;
QUANTITY Vinp ACROSS linp         THROUGH INP      TO electrical_ground;
QUANTITY Vinn ACROSS linn         THROUGH INN      TO electrical_ground;
QUANTITY Vout ACROSS lrou, lcout  THROUGH OUTP   TO OUTN;

BEGIN

linp == Cin*Vinp'DOT;
linn == Cin*Vinn'DOT;
lin  == (Cin/2.0)*Vin'dot;
lcout == Cout *Vout'DOT;
Vout == Rout *lrou;

  M : ENTITY MODELE (behaviour_kernel)
  GENERIC MAP ( noise =>noise, Tsmp=>Tsmp)
  PORT MAP (   VDD=>VDD, VSS=>VSS, MC=>MC, lalim=>lalim,
              INP=>INP, INN=>INN, OUTP=>OUTP, OUTN=>OUTN,
              Commande => Commande);

END ARCHITECTURE behaviour_Zin_out;

----- KERNEL -----

ARCHITECTURE behaviour_kernel OF MODELE IS

BEGIN

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

END ARCHITECTURE behaviour_kernel;

-----

```

nécessite l'introduction de deux nouveaux paramètres (Tsmp, noise). Comme nos sources de bruit sont temporelles et que les caractéristiques de bruit sont fréquentielles nous utiliserons une FFT et donc un échantillonnage pour prélever les signaux. Tsmp sera donc égale à la valeur de la période d'échantillonnage.

MISE EN EQUATION DES IMPEDANCE D'ENTREE ET DE SORTIE DU MODELE

NOYAU FONCTIONNEL

Code 4 - 1 : Exemple de modèle complet en VHDL-AMS

N.B. : nous ne détaillerons par la suite que le noyau fonctionnel de nos modèles.

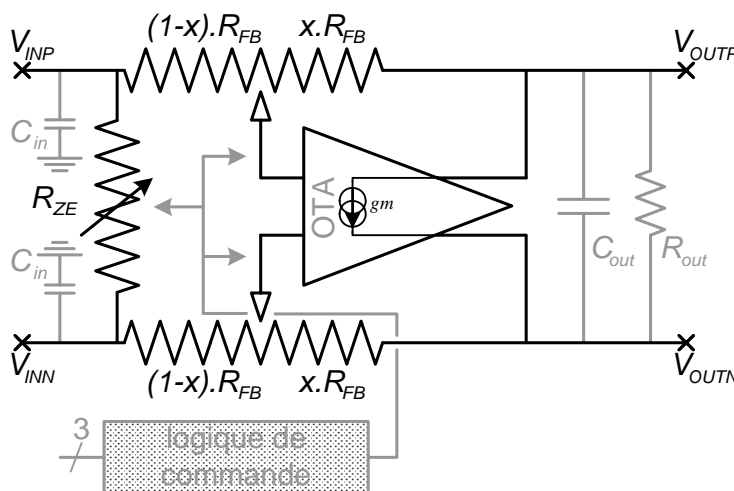
III. Modélisation de la chaîne FI (Fréquence Intermédiaire)

Afin de ne pas surcharger ce chapitre, l'ensemble des modèles en VHDL-AMS est donné en annexe.

III.1. L'amplificateur à Contrôle Automatique de Gain (CAG)

III.1.1. Etude théorique du circuit

L'amplificateur CAG (ou AGC Automatic Gain Control) est placé en début de chaîne de réception. Ce bloc va de paire avec une logique de commande située à la fin de la chaîne de traitement des données formant ainsi une boucle d'asservissement (cf. Figure 4 - 1). Il permet d'adapter l'amplitude du signal d'entrée afin de placer celle-ci dans la plage de fonctionnement des blocs postérieurs à l'amplificateur CAG et d'éviter notamment leur saturation. La Figure 1 - 1 représente le schéma de principe de l'amplificateur. Il est notamment composé d'un amplificateur opérationnel à transconductance (AOT) et de résistances variables commandées par un bloc logique.



$$\text{Gain} = \frac{V_{outp} - V_{outn}}{V_{inp} - V_{inn}} \approx -\frac{(1-x)}{x} \quad \text{Eq. 4 - 1}$$

Figure 4 - 3 : Schéma de l'amplificateur à CAG

Le gain du montage dépend alors de la valeur des résistances $x.R_{FB}$ et $(1-x).R_{FB}$. La valeur de x est déterminée par un code sur trois bits. On a donc potentiellement huit codes différents, seuls six sont utilisés. La différence entre deux gain successifs est de $-6dB$.

III.1.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Seules les caractéristiques principales de l'amplificateur à CAG (gain et impédance d'entrée) ont été modélisées. Le modèle du noyau fonctionnel est disponible en annexe cf. §IV.1. Tous les résultats qui suivent sont obtenus par simulations mixtes, c'est-à-dire associant modèle au niveau transistor et

modèle fonctionnel en VHDL-AMS. Le schéma de test commun aux deux simulations est décrit ci-dessous :

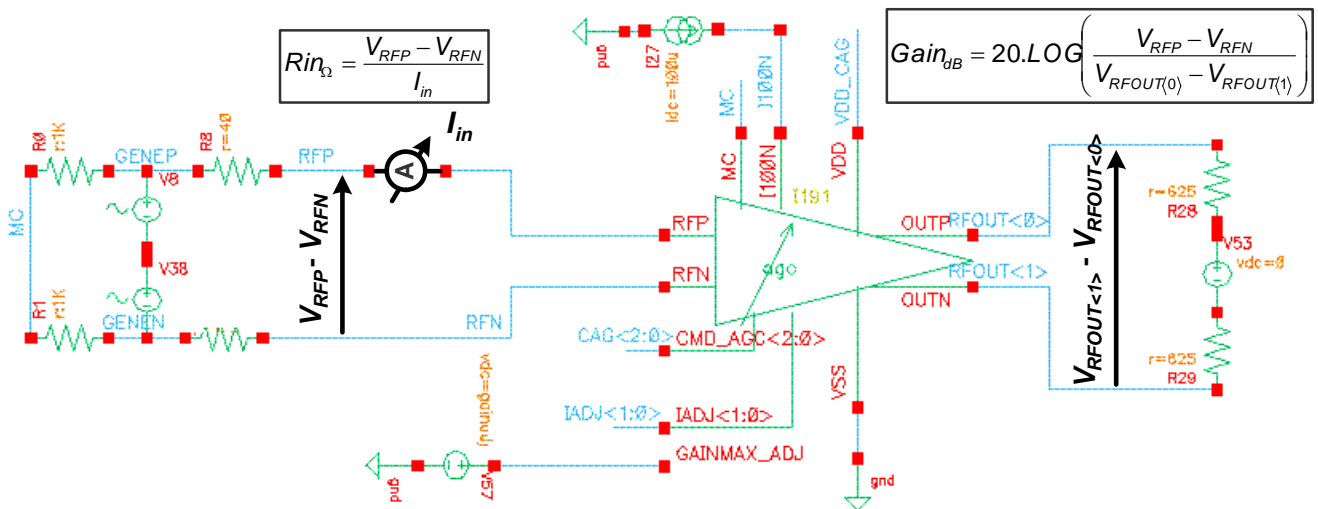


Figure 4 - 4 : Schéma de test du gain et de l'impédance d'entrée de l'amplificateur CAG

Pour obtenir l'impédance d'entrée en fonction des bits de commande, il faut effectuer une analyse fréquentielle (.AC) doublée d'une analyse paramétrique, où les bits de commandes sont incrémentés à chaque passe.

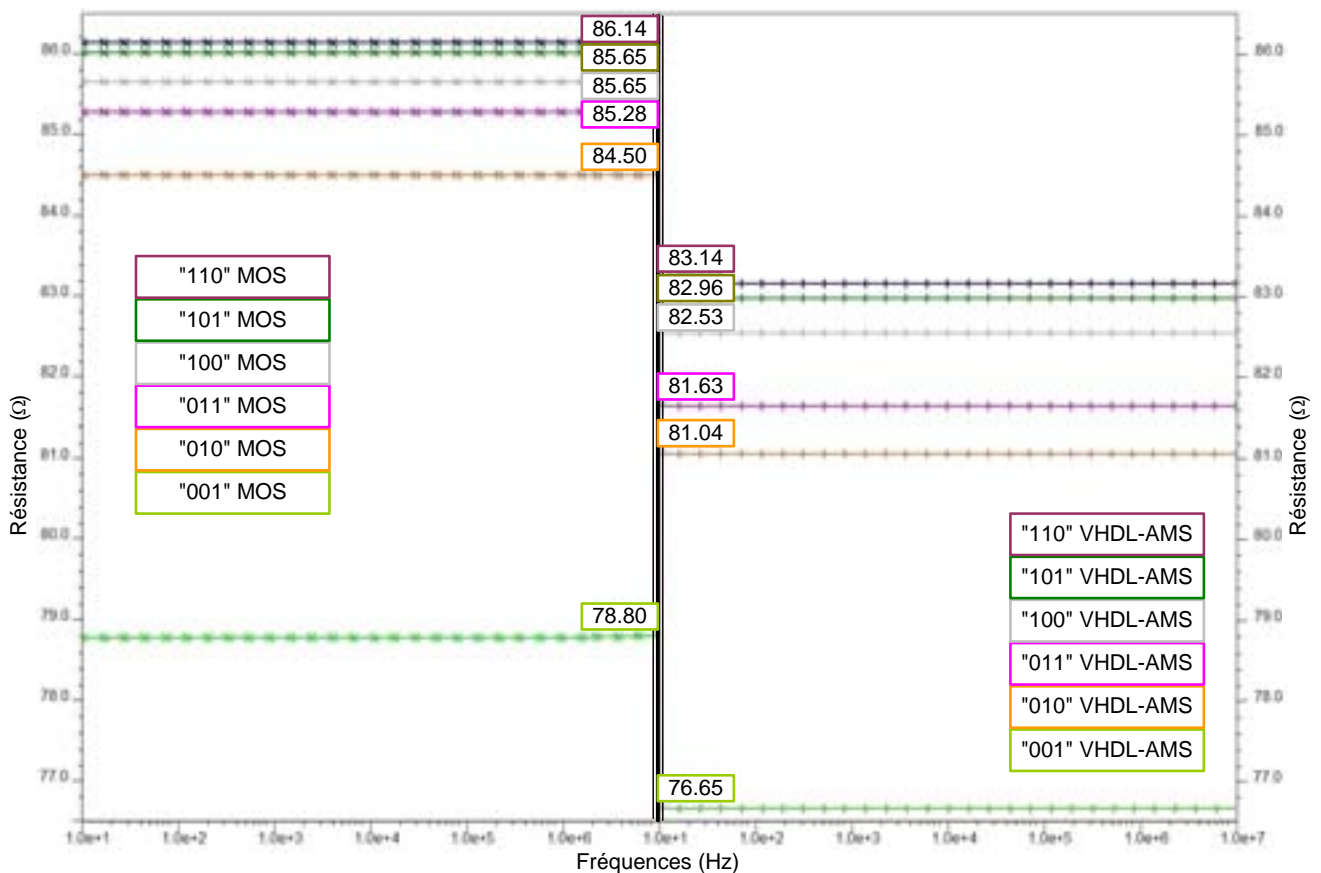


Figure 4 - 5 : Impédances d'entrées du modèle transistor et VHDL-AMS de l'amplificateur CAG

La courbe ci-dessous nous permet de comparer les gains des deux modélisations :

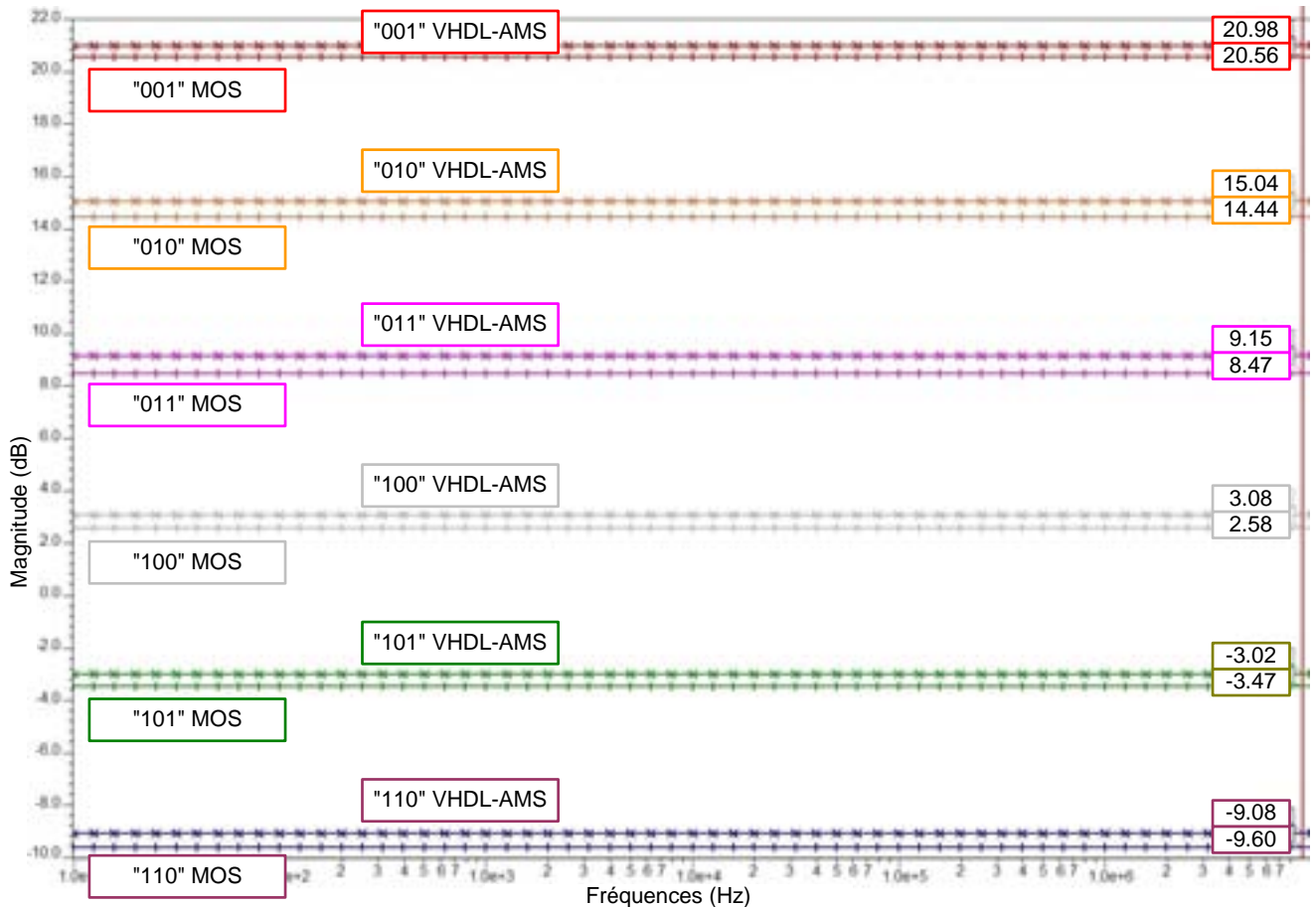


Figure 4 - 6 : Gains des modèles transistor et VHDL-AMS de l'amplificateur CAG

Nous constatons une différence d'au plus 3Ω pour les impédances d'entrée entre la modélisation MOS et VHDL-AMS et d'au plus $0.5dB$ en ce qui concerne le gain.

III.2. Le générateur de quadrature

III.2.1. Etude théorique du circuit

Le générateur de quadrature est utilisé pour réaliser, à partir d'une horloge de fréquence $286MHz$, deux paires d'horloges à $71.5MHz$ déphasées de $\pi/2$ l'une par rapport à l'autre. Les deux composantes de la paire sont en opposition de phase. Le générateur est utilisé conjointement avec deux mélangeurs afin de générer les voies I et Q en quadrature de phase, à partir du signal entrant dans la chaîne de réception RF. Ce générateur est composé de bascules logiques rebouclées qui permettent la division et le déphasage des deux signaux de sortie. La Figure 4 - 7 représente le chronogramme théorique de sortie. Les retards, temps de montée et autres imperfections du générateur ne seront pas pris en compte afin de ne pas surcharger le simulateur.

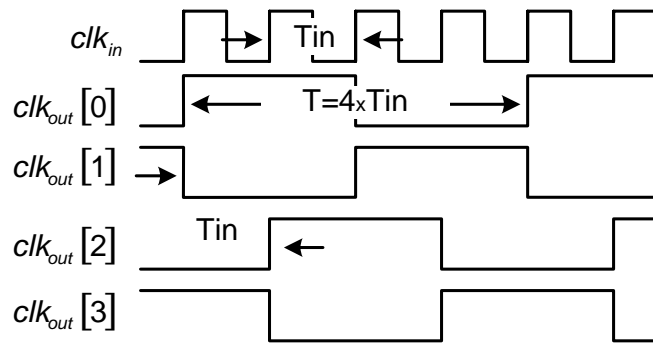


Figure 4 - 7 : Chronogramme du générateur de quadrature

III.2.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Nous réalisons une simulation temporelle mixte afin de comparer les chronogrammes du modèle transistor et du modèle en VHDL-AMS (disponible en annexe cf. §IV.3). Le montage de test, est composé du générateur de quadrature et d'un générateur de signaux à $286MHz$ en entrée.

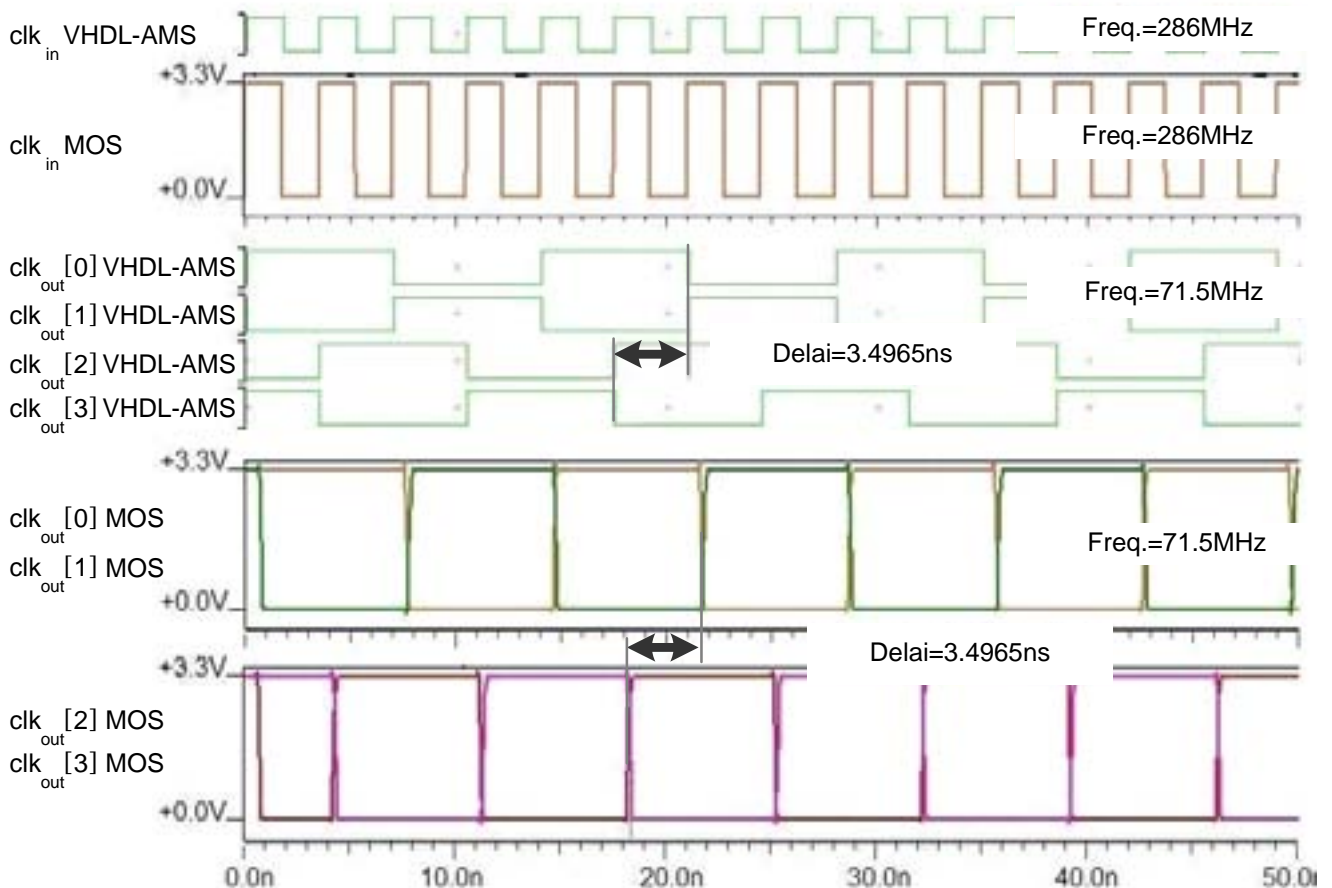


Figure 4 - 8 : Chronogramme des modèles transistor et VHDL-AMS du générateur de quadrature

Nous constatons que, concernant les principales caractéristiques (fréquence de sortie et déphasage), notre modélisation est conforme à la modélisation transistor.

III.3. Le mélangeur

III.3.1. Etude théorique du circuit

Le rôle d'un mélangeur est de traduire un signal d'une fréquence f_1 à une fréquence f_2 . Pour cela le signal est multiplié (ou mélangé) avec celui d'un oscillateur local (O.L.) de fréquence $f_1 + f_2$. Le produit est donc un signal à deux composantes de fréquence f_2 et $2.f_1 + f_2$. Grâce à un jeu de filtres situés après le mélangeur, seule la composante de fréquence f_2 est généralement conservée.

$$A.\sin(2.\pi.f_1.t).\sin(2.\pi.(f_2 + f_1).t) = \frac{A}{2}.\left(\cos(2.\pi.f_2.t) - \cos(2.\pi.(f_2 + 2.f_1).t)\right) \quad \text{Eq. 4 - 2}$$

Dans notre cas, l'oscillateur utilisé est un signal carré fourni par le générateur de quadrature décrit précédemment. Le signal différentiel entrant est alors successivement multiplié par ± 1 (on dit qu'il est haché). Le schéma du mélangeur est représenté ci-dessous :

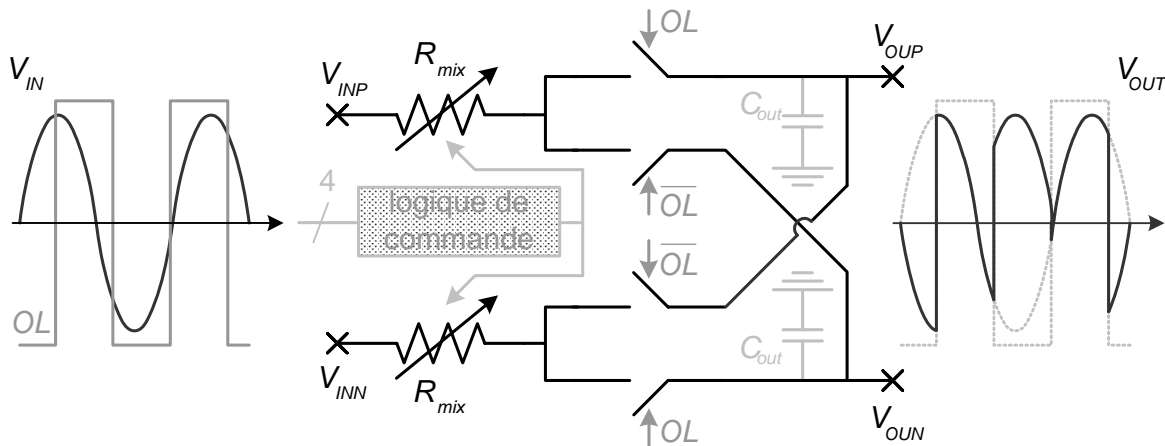


Figure 4 - 9 : Schéma du Mélangeur

L'utilisation d'un signal carré et non sinusoïdal comme oscillateur s'explique par un gain plus important en sortie du mélangeur. Considérons la décomposition en série de Fourier d'un signal carré d'amplitude ± 1 :

$$OL(t) = \frac{4}{\pi} \left[\sin(2.\pi.(f_1 + f_2).t) + \frac{\sin(2.\pi.3.(f_1 + f_2).t)}{3} + \frac{\sin(2.\pi.5.(f_1 + f_2).t)}{5} + \dots \right] \quad \text{Eq. 4 - 3}$$

Nous conservons alors le maximum de puissance du signal initial, puisque le gain est de $2/\pi$ contre $1/2$ si nous avons utilisé une sinusoïde. Soit un gain en dB de $-3.92dB$, contre $-6.02dB$ pour un mélangeur classique.

Pour notre application, la fréquence de l'oscillateur sera de $71.5MHz$ et la fréquence du signal d'entrée de $73.125MHz$, soit une fréquence de sortie (après filtrage) de $1.625MHz$.

III.3.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Nous utilisons le même environnement de test pour le modèle en VHDL-AMS que celui utilisé pour la simulation niveau transistor. Le schéma de test ne comporte qu'un mélangeur (celui générant la voie Q), la voie I est obtenue par le même montage en ajoutant un déphasage de $\pi/2$ à l'oscillateur.

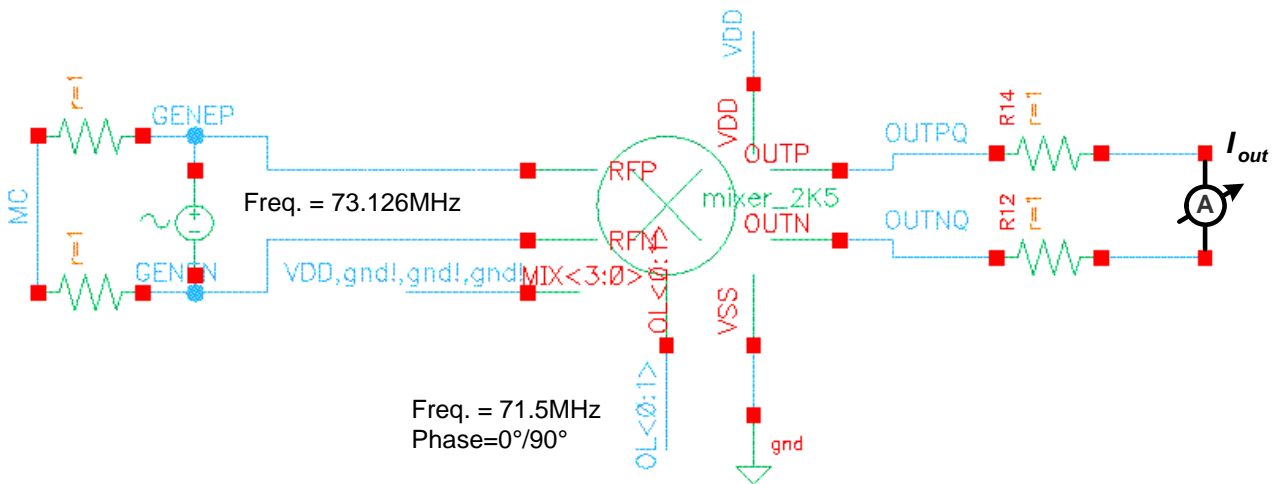


Figure 4 - 10 : Schéma de test du Mélangeur

En simulant simultanément les mélangeurs des voies I et Q, nous obtenons effectivement deux signaux déphasés de $\pi/2$ (I et Q). On distingue également les produits de mélange à 1.625MHz :

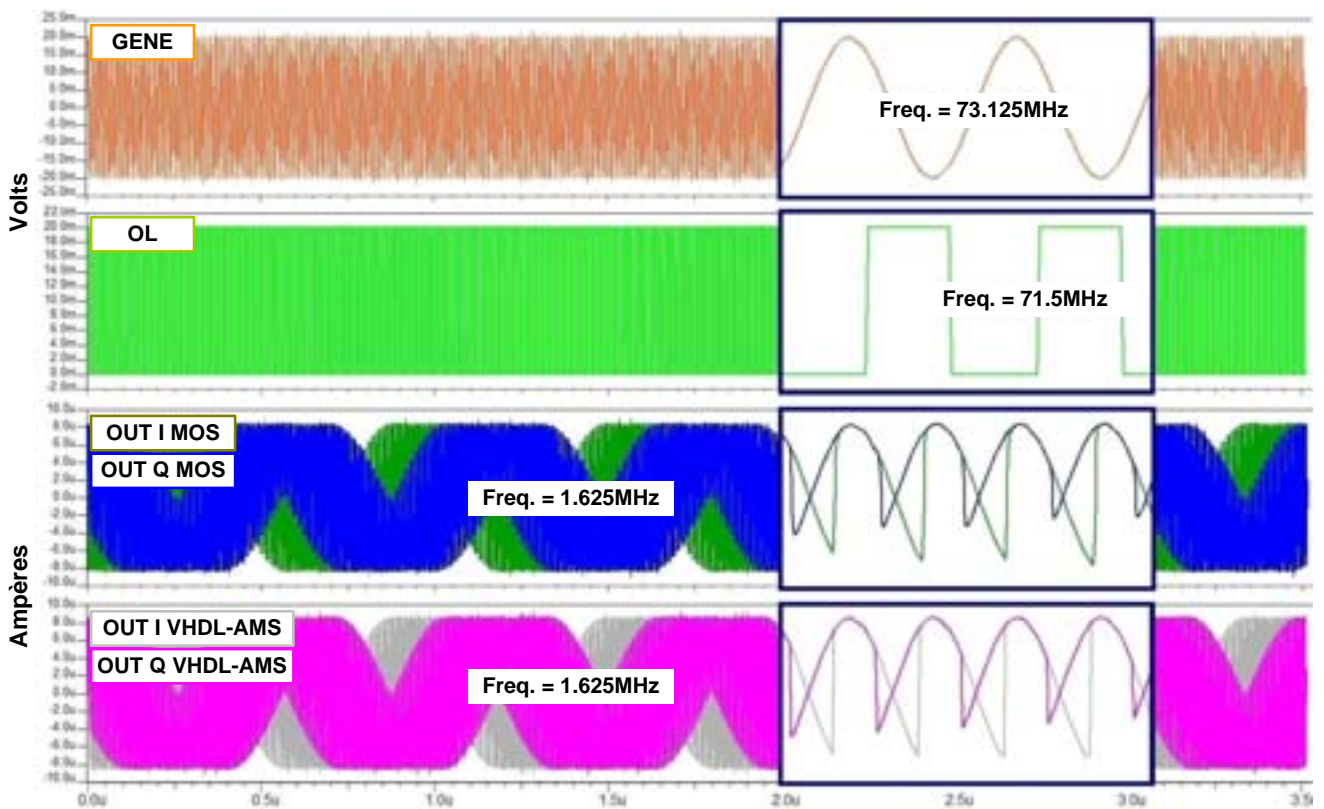


Figure 4 - 11 : Entrées et Sorties du Mélangeur (voies I et Q)

III.4. Le filtre anti-repliements complexe (Anti-Aliasing Filter)

III.4.1. Etude théorique du circuit

Ce bloc est positionné à la suite des deux mélangeurs (I&Q) et avant le convertisseur analogique/numérique. La fréquence d'échantillonnage du modulateur Sigma Delta utilisée pour cette conversion est de $F_{ech} = 13MHz$. Comme illustré par la figure ci-dessous, cet échantillonnage entraîne le repliement des composantes indésirables aux fréquences $f_u \pm n.F_{ech}$ $n \in [1..\infty]$ ainsi que $-f_u$ sur la bande utile centrée à $f_u = 1.625MHz$. Or, une trop grande puissance à cette fréquence peut saturer le modulateur Sigma Delta et n'est donc pas souhaitable.

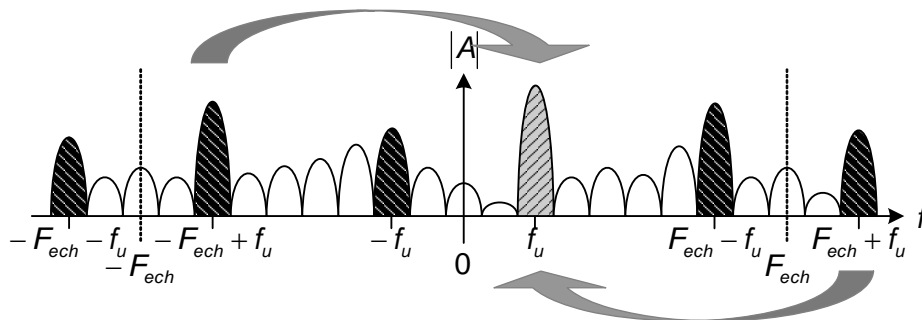


Figure 4 - 12 : Repliement des raies parasites suite à un échantillonnage

Le rôle du filtre anti-repliements est donc d'atténuer ces perturbateurs afin de limiter leur effet lors du repliement. Le fait de traiter simultanément les voies I et Q lui confère le nom de filtre complexe.

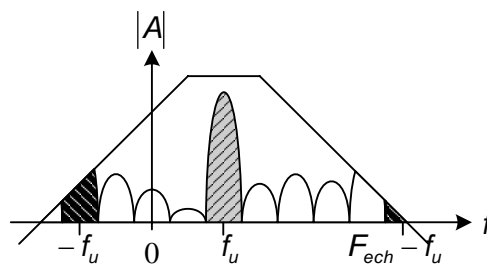


Figure 4 - 13 : Gabarit du filtre anti-repliements

La méthode utilisée pour obtenir un gabarit non symétrique par rapport à l'axe $f = 0$ est celle décrite dans l'Annexe II du Chapitre I (cf. §VII.2.). La dispersion de la valeur des résistances est de l'ordre de 70%. Comme les résistances déterminent la fréquence centrale du filtre, elles sont variables afin de pouvoir régler la fréquence utile une fois le circuit fondu. La commande de la valeur s'effectue par 3 bits. Cette structure développée par M. ROBBE et H. GUEGNAUD n'ayant pas été protégée par un brevet nous ne pouvons être plus précis dans sa description.

III.4.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Le modèle en VHDL-AMS est disponible en annexe cf. §IV.4. Nous utilisons de nouveau le même environnement de test pour ce modèle que celui utilisé pour la simulation au niveau transistor :

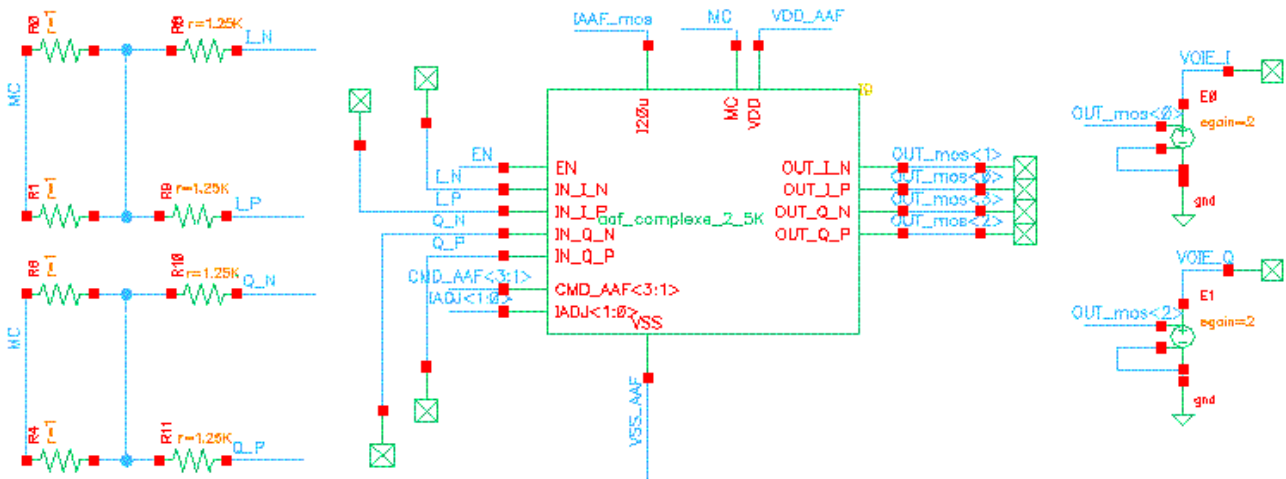


Figure 4 - 14 : Schéma de test du filtre anti-aliasing

Les sources de tension par des voies I&Q sont déphasées de $\pi/2$. Afin de reproduire les conditions réelles d'utilisation, l'impédance d'entrée des générateurs de tension est de $1.25k\Omega$ ce qui correspond à l'impédance de sortie des mélangeurs qui sont placés en amont du filtre. Nous effectuons alors une étude fréquentielle pour comparer le gabarit du filtre des deux modélisations :

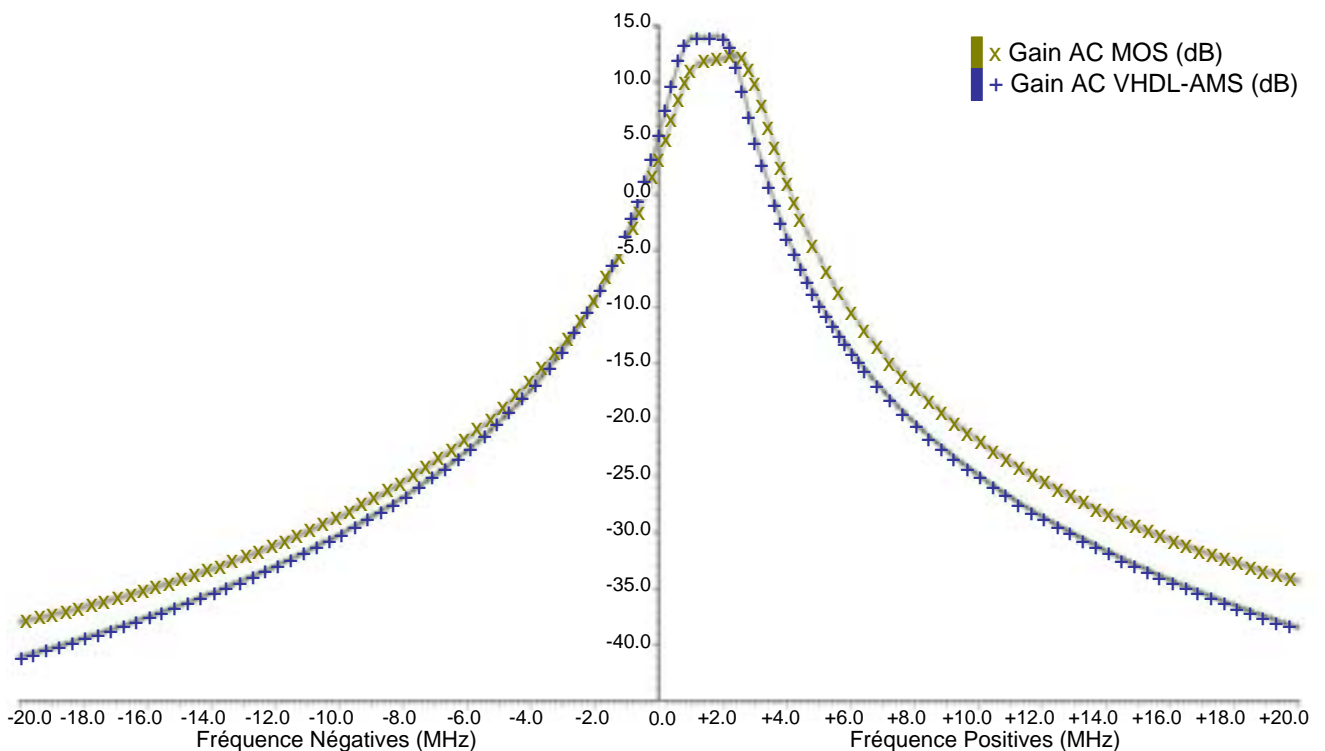


Figure 4 - 15 : Analyse AC des deux modèles de filtre

III.5. La boucle à verrouillage de phase (PLL)

III.5.1. Etude théorique du circuit

Comme il n'existe pas de quartz à la fréquence de 286MHz , et comme il est plus économique de générer en interne la fréquence à partir d'une référence disponible dans le commerce (quartz 26MHz utilisé dans les GSM, donc à faible coût), nous utilisons une PLL (Phase Locked Loop) comme source de fréquences du générateur de quadrature. Comme décrit ci-dessous (cf. Figure 4 - 16), la PLL est composée d'un Comparateur de Phase, d'un Filtre de Boucle, d'un oscillateur contrôlé en tension à réseau LC externe (Voltage Control Oscillator) et d'un diviseur de fréquence par 22. Cette boucle asservit la sortie du VCO divisée par 22 à une référence de fréquence. On obtient alors en sortie du VCO une source de fréquence à 286MHz .

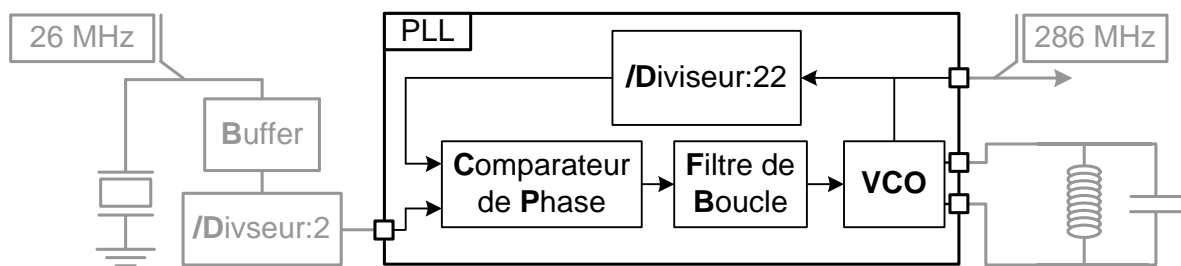


Figure 4 - 16 : Schéma bloc de la PLL

Le montage et les résultats de mesures que nous allons présenter dans cette sous-partie sont issus d'une collaboration avec Thierry LAGUTERE (thèse convention CIFRE EADS/LAII soutenue le 24-03-05). Ainsi, contrairement aux précédents blocs, la PLL ayant fait l'objet de travaux plus approfondis, nous avons eu accès aux mesures des principales caractéristiques de la PLL : la fréquence de sortie et le bruit de phase.

La finalité de notre travail n'est pas l'étude détaillée de la PLL mais l'obtention d'un modèle complet en VHDL-AMS de la chaîne rapide à simuler. Nous avons donc réalisé une modélisation globale de la PLL (par opposition à une modélisation bloc à bloc). En utilisant les résultats de simulation et de mesure, nous avons réalisé une approximation du bruit de phase de la PLL par trois tangentes : un bruit en $1/f$ jusqu'à 3kHz , puis un bruit blanc ($L(f) = 122\text{dB}_c/\text{Hz} \Rightarrow \text{DSP}(f) = 125\text{dB}_c/\text{Hz}$) jusqu'à 300kHz , puis un bruit en $1/f^2$ au-delà de cette fréquence. Pour réaliser cette forme de bruit, nous avons eu recours aux modèles de bruit développés dans le Chapitre II.

III.5.2. Simulations mixtes (modèle transistor / VHDL-AMS)

La figure, présentée ci-dessous, nous permet de comparer le bruit de phase de notre modèle en VHDL-AMS aux mesures :

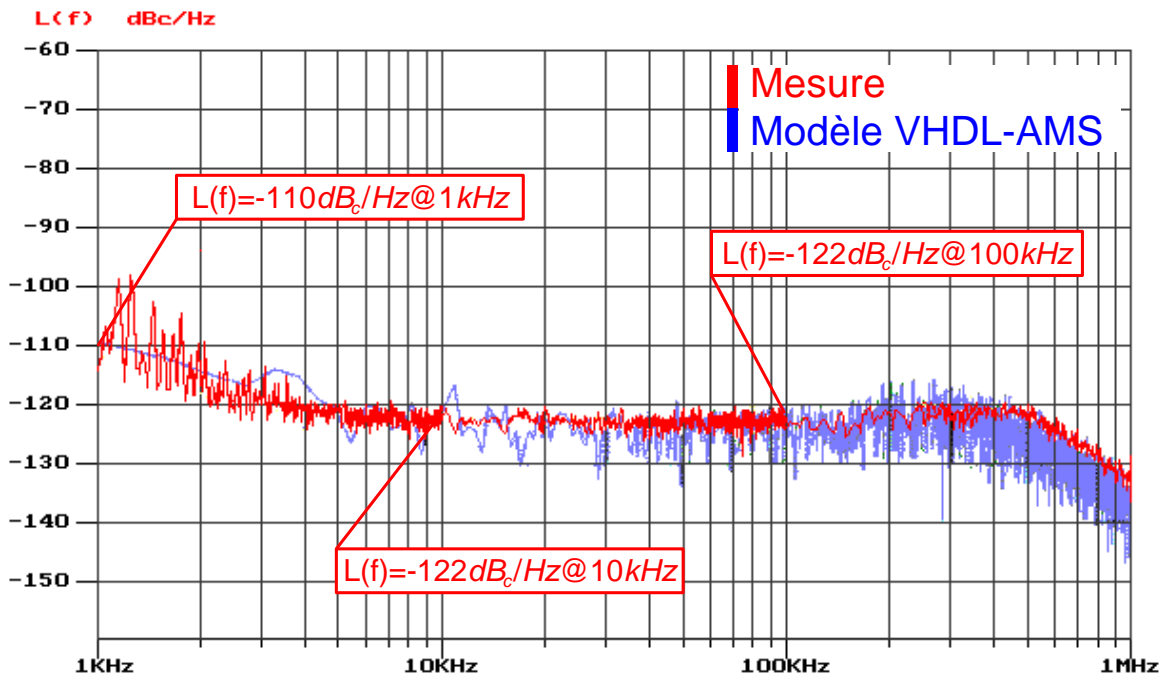


Figure 4 - 17 : Comparaison des bruits de phase des modèles de la PLL et de la mesure

Notre modèle est moins précis que la mesure, (il ne reproduit pas la bosse autour de la fréquence 300kHz), sa simulation est cependant très rapide (~7 minutes).

III.6. La chaîne FI

Dans un premier temps, nous avons réalisé le sous-ensemble composé de l'amplificateur CAG, du générateur de quadrature, des deux mélangeurs et du filtre anti-repliements. Nous disposons, grâce aux simulations au niveau transistor, d'une estimation des paramètres de bruit des blocs concernés :

CAG	Mélangeur	Filtre Antirepliements
$0.9nV / \sqrt{Hz}$	<i>négligeable</i>	$10.6nV / \sqrt{Hz} / \text{voie}$

Tab. 4 - 1

Pour vérifier le bon fonctionnement du sous-ensemble, nous réalisons une analyse transitoire suivie d'une analyse fréquentielle (périodogramme). Pour mettre en valeur le fonctionnement du filtre anti-repliement dans le domaine fréquentiel, notre signal d'entrée sera composé de quatre signaux aux fréquences : $f_i \pm f_u$ et $f_i \pm (f_u + f_{sd})$. Pour notre application, f_i désigne la fréquence intermédiaire $f_i = 71.5MHz$, f_{sd} est la fréquence d'échantillonnage du modulateur Sigma Delta $f_{sd} = 13MHz$ et

f_u désigne la fréquence utile $f_u = 1.625\text{MHz}$ soit $1/8$ de la fréquence d'échantillonnage du modulateur.

Une fois amplifié par l'amplificateur CAG, le signal est ramené autour de la fréquence nulle par les deux mélangeurs en quadrature de phase. Nous obtenons alors (notation complexe) huit raies aux fréquences : $\pm f_u$, $\pm(f_u + f_{sd})$, $2 \times f_i \pm f_u$ et $2 \times f_i \pm (f_u + f_{sd})$. Ce signal est ensuite filtré par le filtre anti-repliements. La figure suivante présente le résultat de la simulation. Pour obtenir une analyse de Fourier fiable nous prélevons un grand nombre d'échantillons ($N = 131072$) et effectuons un périodogramme avec un fenêtrage rectangulaire, sur des fenêtres de $NN = 8192$ entrelacées à 50% :

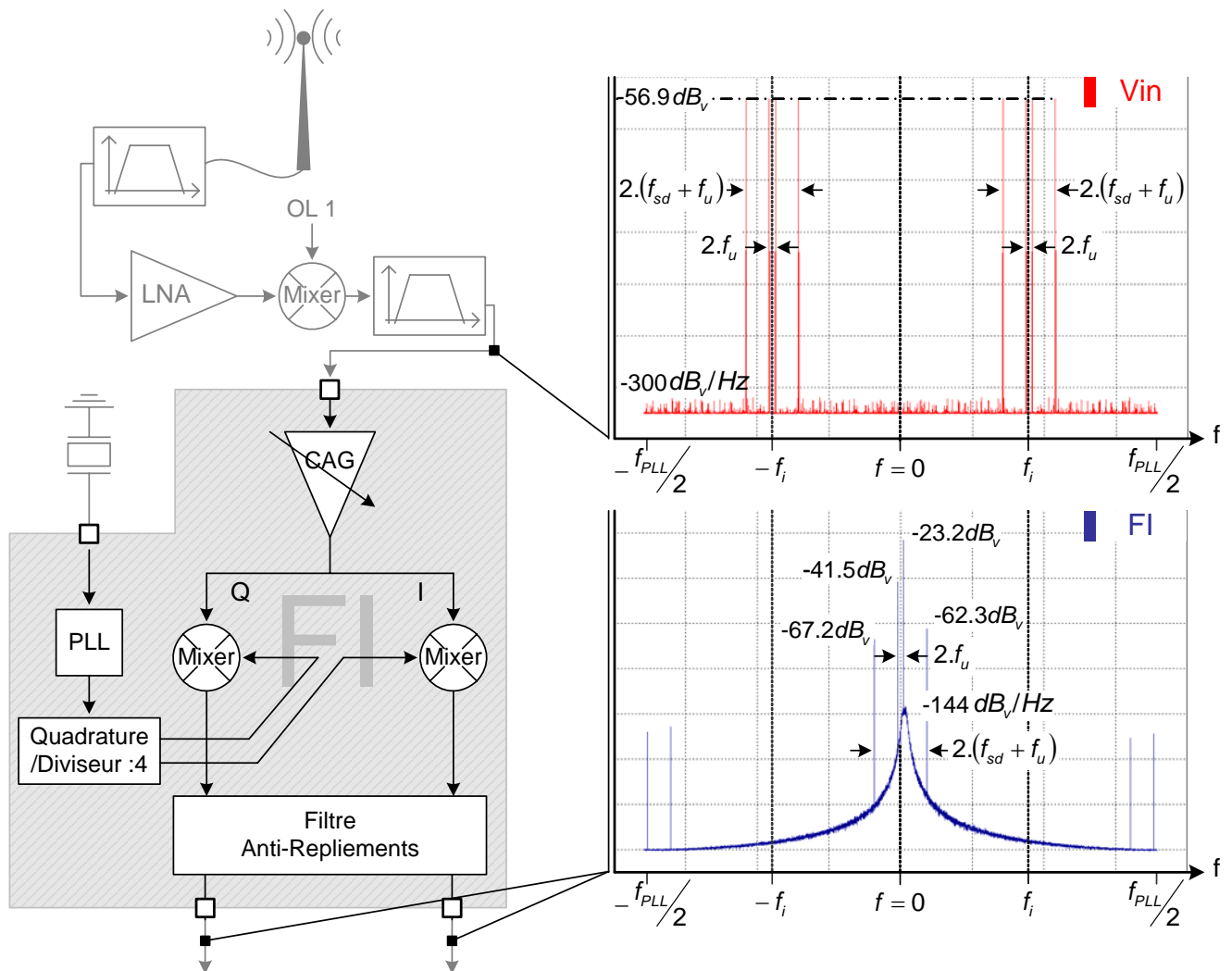


Figure 4 - 18 : Simulation du bloc FI

Le périodogramme de sortie est conforme à la théorie et aux études de chacun des blocs pris séparément. Le gain appliqué au signal sur la bande utile est de $33.6dB$, le gain théorique (qui est également celui mesuré sur la puce) est de :

$$32.5dB = \underbrace{21dB}_{\text{Gain th CAG}} + \underbrace{(-4 + 3)dB}_{\text{Gain th 2 Mixers}} + \underbrace{12.5dB}_{\text{Gain th Filtre}} \quad \text{Eq. 4 - 4}$$

Cette différence de gain était prévisible car le gain de notre modèle de filtre anti-repliements est supérieur au gain théorique (cf. Figure 4 - 15). Les raies parasites sont également fortement diminuées par le filtre anti-repliements.

La forme du bruit en sortie du filtre anti-repliements est conforme au gabarit du filtre. La valeur de la Densité Spectrale de Puissance du bruit thermique mesurée sur la puce est de : $-143.7dB_v/Hz$, la valeur de la DSP bruit thermique simulée (cf. Figure 4 - 18) est de $-144dB_v/Hz$ dans la bande utile $[f_u - 50kHz, f_u + 50kHz]$ ce qui est conforme à nos calculs :

$$DSP_{\text{thermique FI}} = 20 \cdot \text{Log}_{10} \left(\sqrt{\left(0.9e^{-9} \times 10^{\frac{21}{20}} \times 10^{\frac{-4+3}{20}} \right)^2 + \left(10.6e^{-9} \right)^2} \times 10^{\frac{13.5}{20}} \right) \quad \text{Eq. 4 - 5}$$

$$\approx -143.6dB_v/Hz$$

N.B. : l'erreur sur le gain du filtre anti-repliements a compensé l'erreur d'estimation du bruit thermique de chacun des blocs.

Une fois ce sous-ensemble vérifié, nous avons ajouté la PLL et de nouveau effectué une analyse temporelle suivie d'une analyse fréquentielle. Les résultats de cette simulation n'ont pas été concluants. Il semble que le simulateur ait rencontré quelques difficultés à générer le bruit de phase de la PLL. Le noyau de simulation numérique (qui gère les horloges de commande des mélangeurs) et le noyau de simulation analogique (qui gère le reste de la chaîne) rencontrent des problèmes de synchronisation. L'utilisation de la fonction *BREAK* dédiée à la synchronisation n'y a rien changé. Après de nombreuses autres tentatives, nous restons à ce jour dans l'expectative. Néanmoins, nous verrons dans la troisième partie du chapitre comment nous avons contourné cette difficulté pour obtenir la contribution de la PLL dans des performances globales de la chaîne de réception RF.

Maintenant que nous disposons d'un modèle fiable de la chaîne FI, intéressons nous au fonctionnement du modulateur Sigma Delta Complexe ainsi qu'à la modélisation de ses principales fonctionnalités.

IV. Modélisation du modulateur Sigma Delta

Ce circuit, dédié à la conversion analogique/numérique des données issues de la chaîne FI, est composé d'un générateur d'horloges non-recouvrantes, de deux CAN, de deux CNA et d'un filtre résonateur complexe (qui traite simultanément les voie I et Q).

IV.1. Le générateur d'horloges non-recouvrantes

IV.1.1. Etude théorique du circuit

Ce bloc fournit, à partir d'une horloge de fréquence $26MHz$, huit horloges complémentées, non recouvrantes ($CK\langle i \rangle$ $i \in [0..7]$) et une horloge décalée $Latch_{CAN}$, toutes de fréquence $13MHz$. Un chronogramme explicatif est fourni ci-dessous :

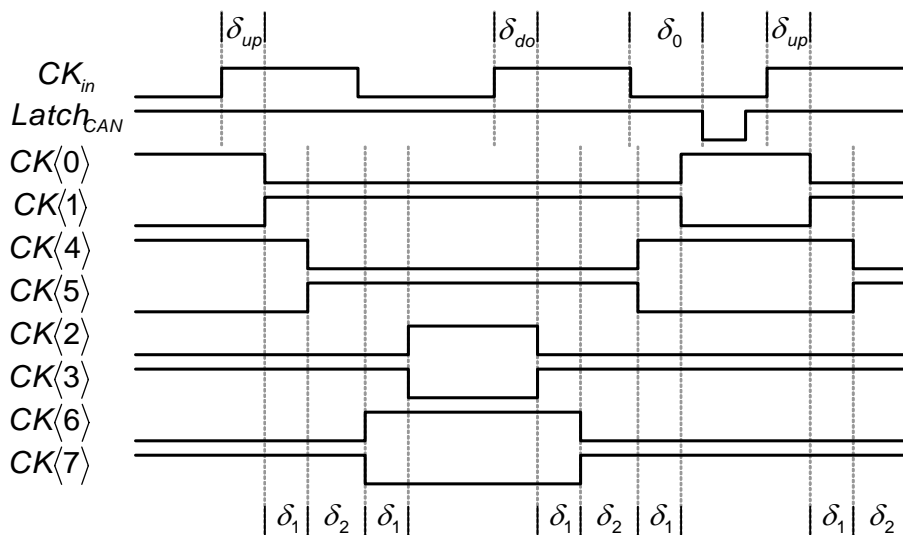


Figure 4 - 19 : Chronogramme de la répartition des horloges

Notre modèle reprend la chronologie décrite par la figure ci-dessus et est disponible en Annexe (cf. §IV.7). Afin de tenir compte du jitter généré par ce bloc (cf. Chapitre III), nous avons ajouté un aléa de répartition gaussienne dans les délais de commutation δ_{up} et δ_{do} . Ce décalage aléatoire influence l'ensemble des signaux puisque les délais δ_0 , δ_1 et δ_2 sont décomptés à partir de δ_{up} et δ_{do} .

IV.1.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Le montage de test est composé d'un générateur de fréquence à $26MHz$ et du générateur d'horloges non-recouvrantes.

La courbe ci-dessous, issue de cette simulation mixte, représente les horloges paires ; les horloges impaires sont obtenues en prenant le complémentaire :

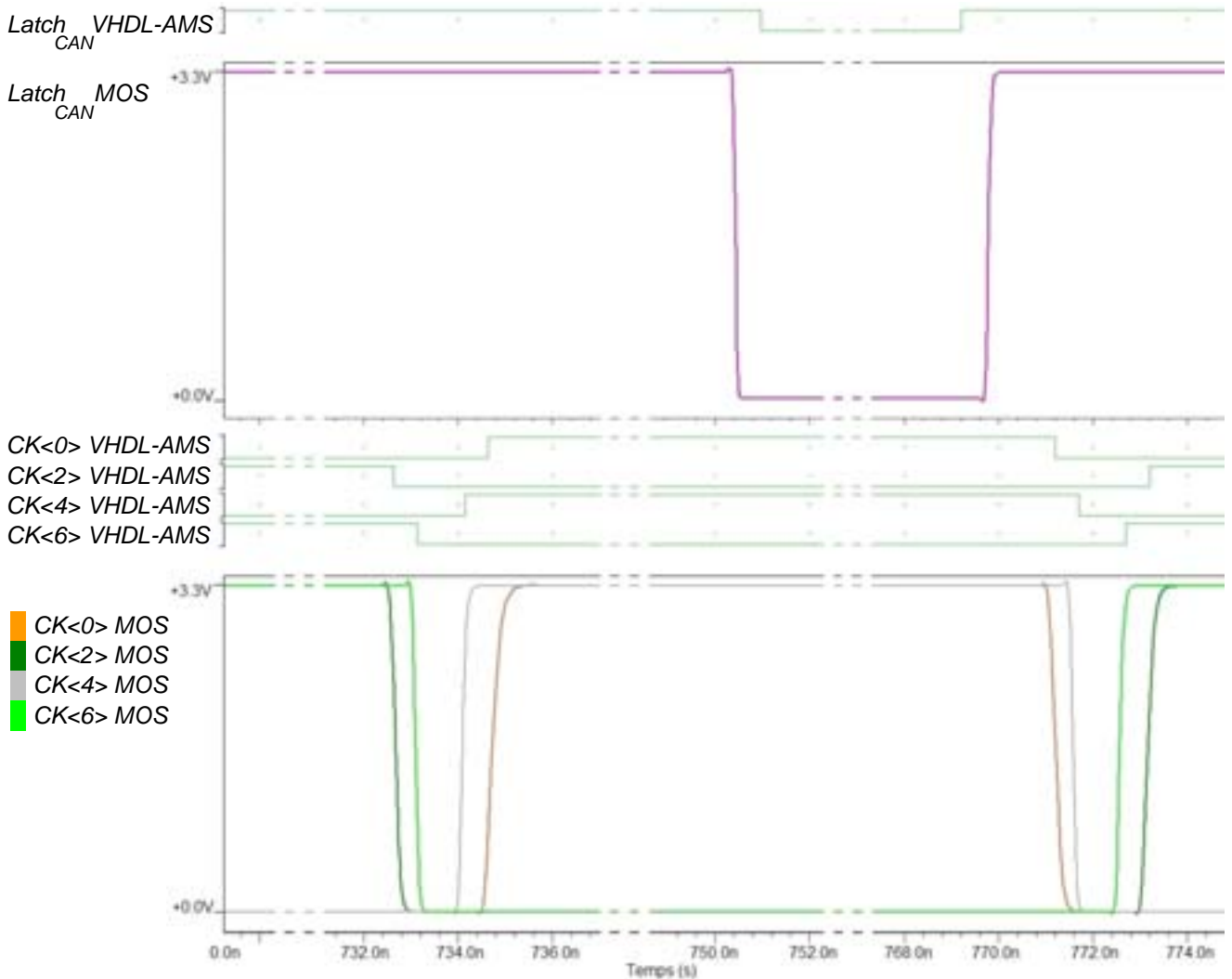


Figure 4 - 20 : Chronogramme des modèle transistor et VHDL-AMS du générateur d'horloges

IV.2. Le Convertisseur Analogique/Numérique

IV.2.1. Etude théorique du circuit

Le convertisseur analogique/numérique que nous utilisons dans le modulateur Sigma Delta est composé de deux sous circuits : un quantificateur, qui permet la numérisation du signal analogique échantillonné sur front descendant du Latch, et un encodeur logique. Le quantificateur délivre un signal de type thermomètre qui est ensuite transformé en code binaire par le second sous circuit.

Le principe de fonctionnement du CAN à entrée différentielle est décrit à travers la Figure 4 - 21. Les résistances montées en série forment une échelle de tensions de comparaison. Des comparateurs

différentiels effectuent la comparaison entre la valeur entrant et les multiples du pas fournissant ainsi le code thermomètre. Cette comparaison est déclenchée par un front descendant du Latch :

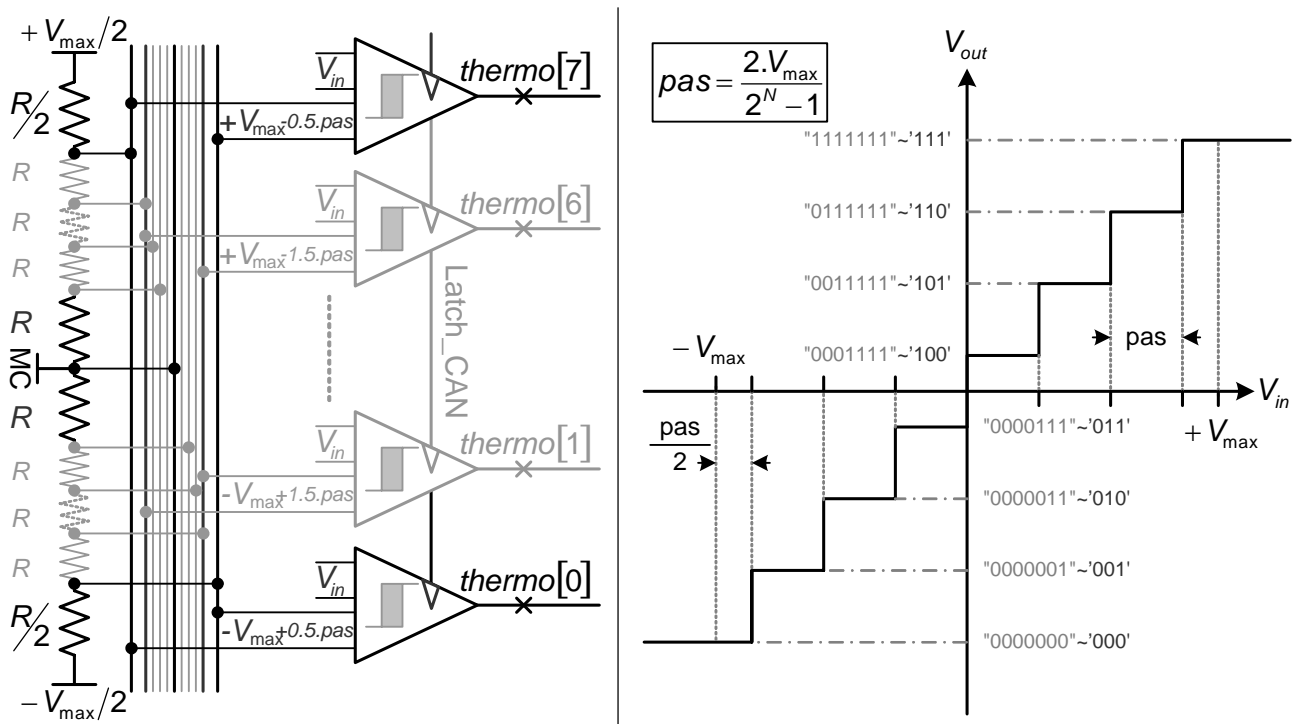


Figure 4 - 21 : Schéma de principe et fonction de transfert d'un Quantificateur 7 niveaux

N.B. : Contrairement à ce nous avons représenté ci-dessus, par manque de place, nous utilisons en réalité deux CAN avec un code thermomètre sur 15 niveaux et un code binaire sur 4 bits. Le pas de conversion est égale à : $pas \approx 46.37mV$ soit $V_{max} \approx 348mV$.

IV.2.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Pour modéliser le quantificateur, nous nous sommes placés dans le cas idéal, c'est à dire lorsque le passage d'un code à un autre s'effectue de manière régulière (pas identiques). Nous avons également considéré que le temps de traitement était constant quel que soit le code. Le modèle du CAN est alors composé du modèle du quantificateur ayant un pas de comparaison et un retard constants comme paramètres d'entrées et du modèle de l'encodeur numérique qui génère un code binaire non signé à partir du code thermomètre. Comme les entrées du CAN sont échantillonnées (sorties du filtre à capacité commutées) et que les sorties sont numériques, nous avons eu, tout naturellement, recours à des entrées/sortie de type *SIGNAL* pour modéliser les codes thermomètre et binaire. Les deux modèles sont disponibles en Annexe (cf. §0).

Pour tester simultanément les deux modélisations, nous appliquons une rampe variant de $-400mV$ à $+400mV$ aux bornes d'entrées des deux modèles afin d'exciter tous les codes. Comme l'entrée du modèle en VHDL-AMS est également de type *SIGNAL*, nous avons eu recours à un échantillonneur déclenché par le latch en amont du modèle. On obtient alors les courbes suivantes :

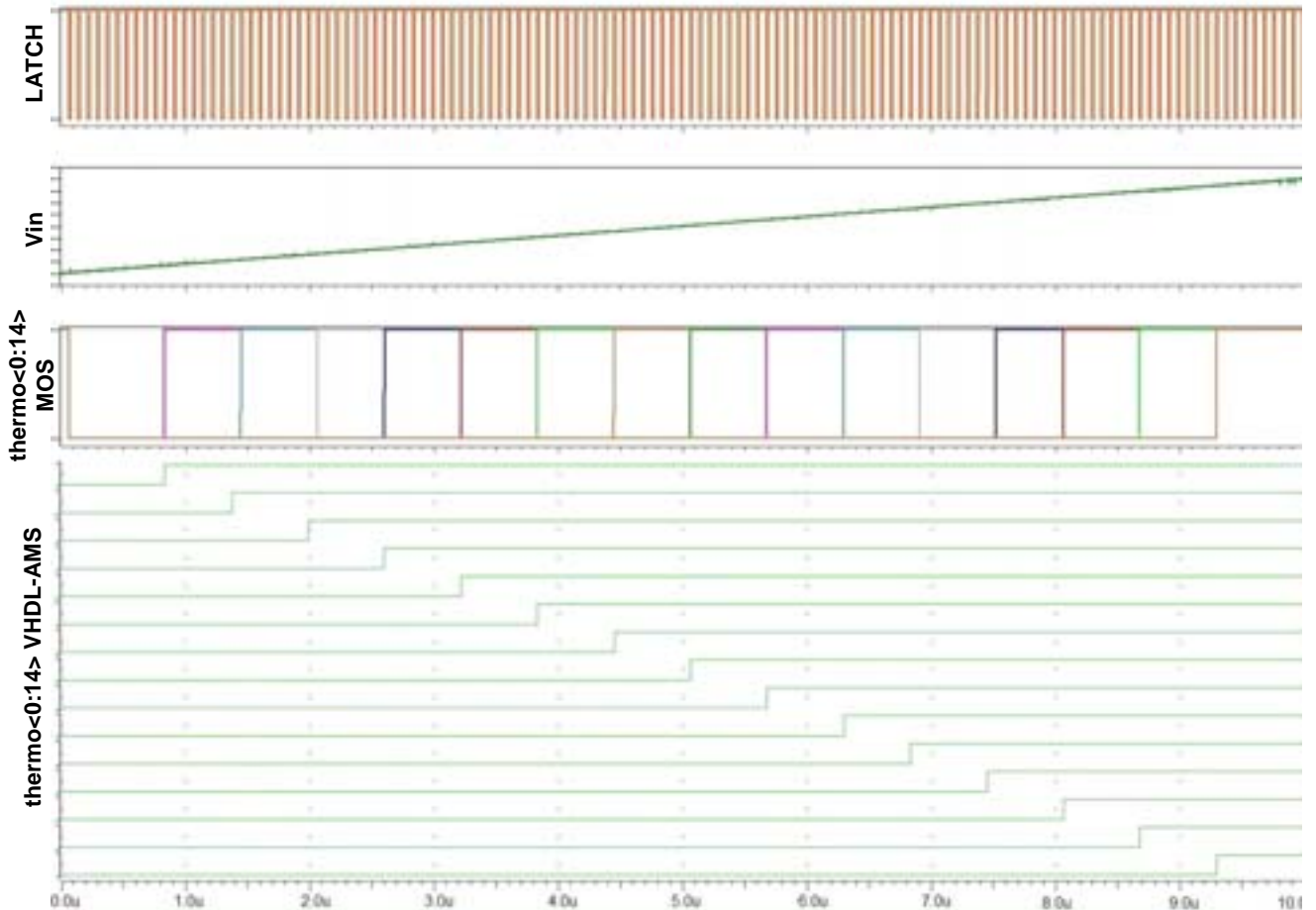


Figure 4 - 22 : Codes thermomètres des deux modèles

Notre modèle de CAN reprend bien les principales caractéristiques du circuit de conversion.

IV.3. Le Convertisseur Numérique/Analogique

IV.3.1. Etude théorique du circuit

Le CNA utilise des sources de courant montées en parallèle qui délivrent dans un sens ou dans l'autre, suivant la configuration du bit de commande, un courant I_{REF} . Ces courants sont alors sommés et traversent une paire de résistances qui génèrent alors une tension différentielle (cf. Figure 4 - 23). Comme pour le CAN, précédemment décrit, seul le comportement idéal (sans imperfection de pas, ni décalage de la tension de mode commun) du CNA sera pris en compte :

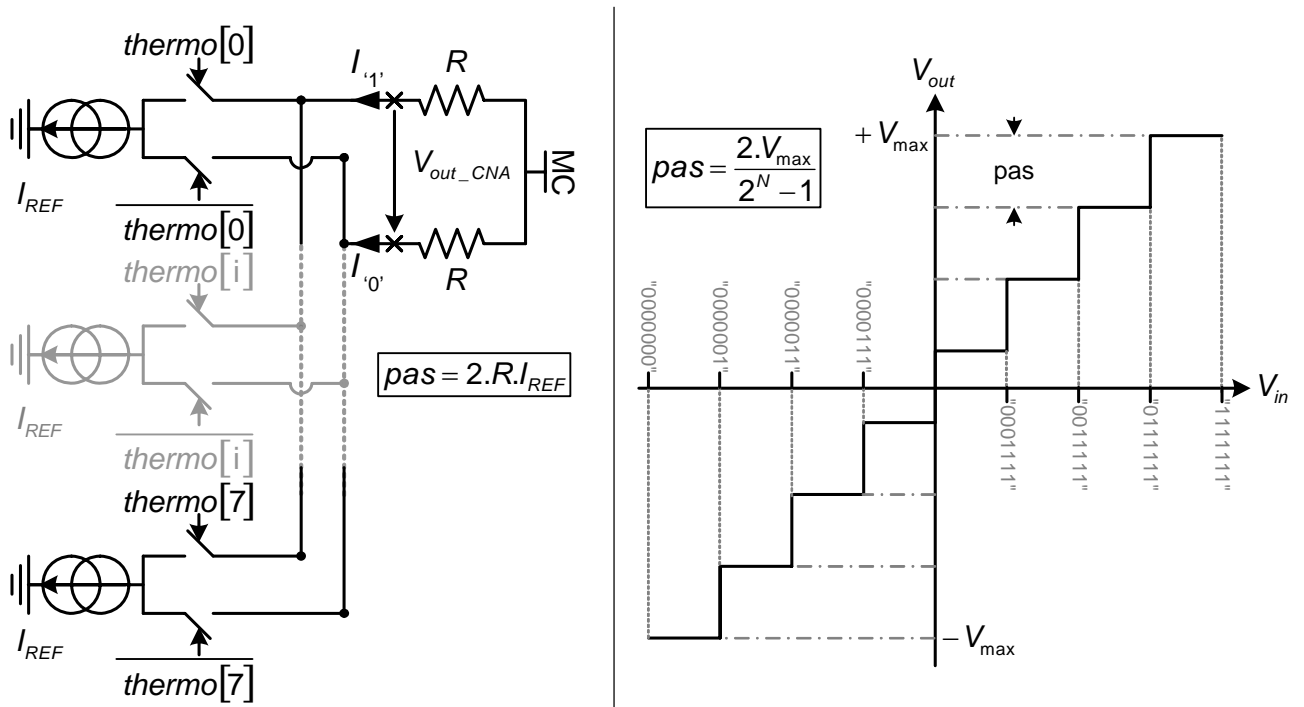


Figure 4 - 23 : Schéma de principe et fonction de transfert d'un CNA 7 bits

N.B. : Comme pour le quantificateur nous n'avons pas pu représenter, par manque de place, le CNA avec un code thermomètre sur 15 bits qui est utilisé dans la structure du Sigma Delta. Le pas de conversion est égale à : $pas \approx 53.33mV$ soit $V_{max} = 400mV$.

Conformément à ce qui est représenté ci-dessus, la tension de sortie est égale à :

$$V_{out_CNA} = R \cdot I_{REF} \cdot \left(\sum (thermo = '1') - \sum (thermo = '0') \right) \quad \text{Eq. 4 - 6}$$

IV.3.2. Simulations mixtes (modèle transistor / VHDL-AMS)

Pour obtenir les courbes ci-dessous, nous branchons les deux modèles de CNA à la suite du CAN. Nous effectuons ensuite une simulation temporelle. Les deux réponses sont identiques :

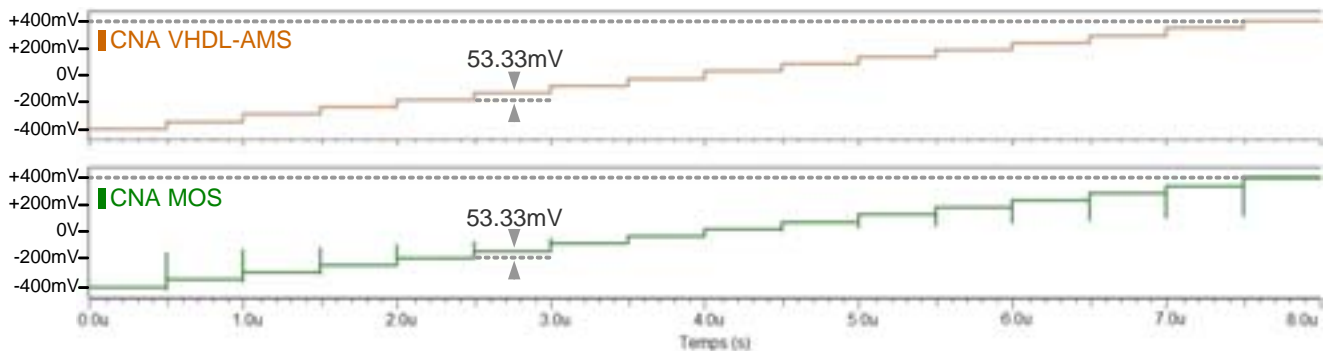


Figure 4 - 24 : Simulation temporelle des deux modélisations du CNA

IV.4. Le résonateur

IV.4.1. Etude théorique du circuit

Classiquement, les filtres à capacités commutées sont des filtres réels qui traitent séparément la voie I et la voie Q. Toute l'originalité du filtre à capacités commutées que nous utilisons réside dans le traitement simultané des voies I et Q (complexe). Le principe du filtre résonateur complexe est celui décrit dans le Chapitre I (cf. Annexe §VII.2). Il est obtenu par la rotation dans l'espace complexe du gabarit d'un filtre réel. Notre résonateur est centré autour de la fréquence utile $f_u = 1.625\text{MHz}$:

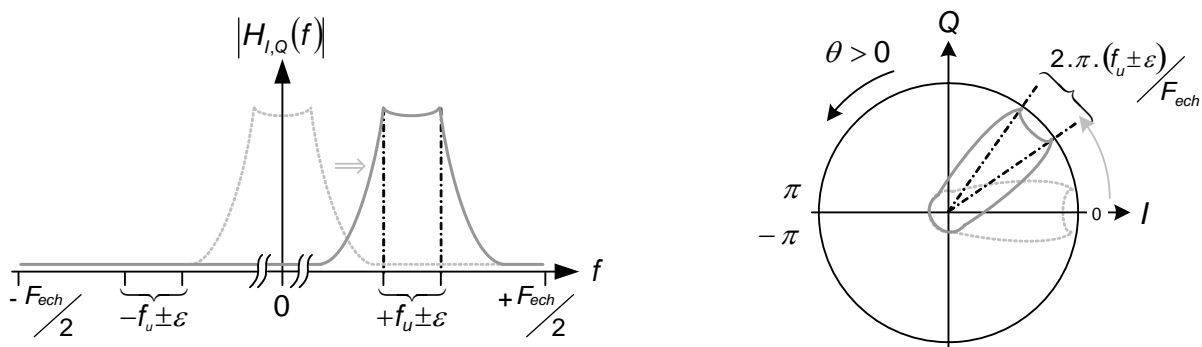


Figure 4 - 25 : Rappel du gabarit du filtre dans le domaine fréquentiel

Comme pour le filtre anti-repliements complexe, cette structure n'est pas encore protégée par un brevet. Nous ne pouvons pas aller plus en avant dans la description de la structure électronique, ni divulguer les équations qui régissent son fonctionnement.

IV.4.2. Simulations VHDL-AMS

Les signaux issus de la chaîne FI étant des signaux analogiques (continus dans le temps et l'amplitude) les entrées du modèle du résonateur seront de type *TERMINAL*. Les sorties ainsi que les entrées issues du CNA étant des signaux échantillonnés, nous utiliserons le type *SIGNAL (REAL)*. Le modèle du résonateur est disponible en Annexe (cf. §IV.10).

Comme une simulation purement fréquentielle (AC) est impossible avec des structures à capacités commutées, nous réalisons une analyse temporelle suivie d'une FFT pour obtenir le comportement en fréquence du filtre complexe. Afin d'obtenir l'ensemble du spectre, nous appliquons un bruit blanc (spectre constant quelque soit la fréquence $PSD = -180\text{dB}_V/\text{Hz}$) à chaque entrée du filtre complexe et nous effectuons un périodogramme à partir de $N = 131072$ échantillons sur des fenêtres de $NN = 16384$ échantillons entrelacées à 50% avec un fenêtrage rectangulaire.

Au regard du grand nombre d'échantillons nécessaires à l'obtention d'un spectre exploitable, nous ne pouvons effectuer de simulation temporelle du modèle transistor. Nous ne présentons donc que le résultat de simulation du modèle VHDL-AMS.

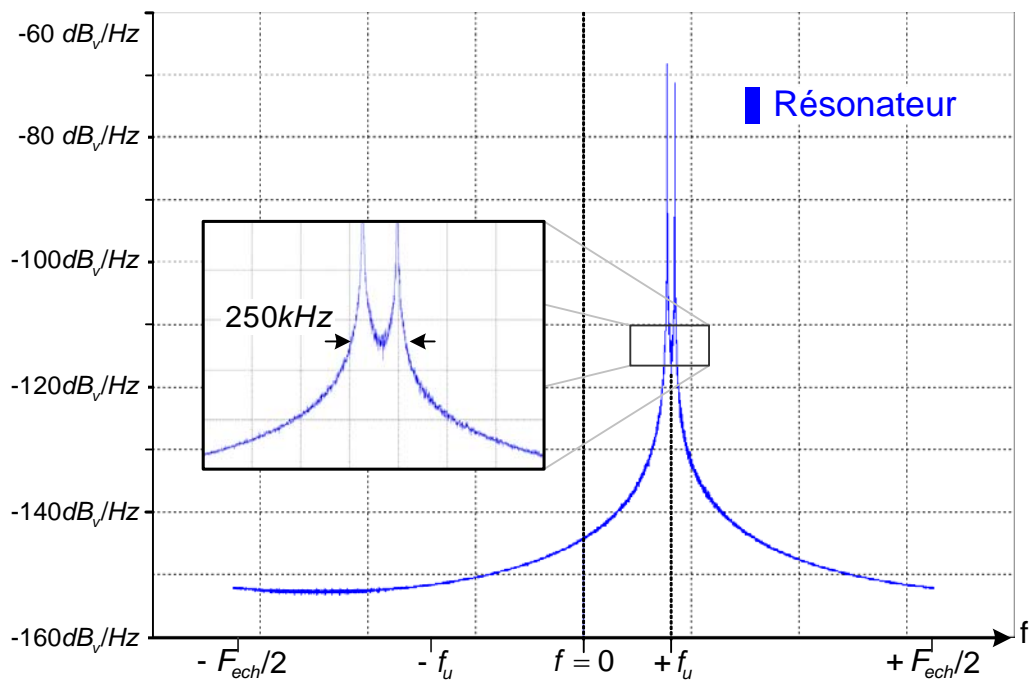


Figure 4 - 26 : Spectre du modèle VHDL-AMS du filtre complexe

Nous retrouvons la dissymétrie du filtre résonateur par rapport à l'axe des y ($f = 0$). Notre résonateur étant composé de deux zéros décalés en fréquence, nous retrouvons cette caractéristique dans le spectre avec deux résonances autour de la fréquence $f_u = 1.625\text{MHz}$.

IV.5. Le modulateur Sigma Delta

IV.5.1. Etude théorique du circuit

Le modulateur utilisé dans la chaîne RX est du second ordre et traite simultanément les voies I et Q issues du filtre anti-repliements. Il comporte deux paires de CAN – CNA, un filtre à capacités commutées complexe (qui fait également office d'échantillonneur et d'additionneur – soustracteur) et un générateur d'horloges non-recouvrantes (destinées à la commande des interrupteurs MOS) et le Latch pour le CAN (cf. Figure 4 - 27).

Nous avons pris le parti de modéliser les blocs constituant le modulateur sans bruit thermique, afin de ne considérer cette imperfection qu'au niveau du Sigma Delta complet. Les simulations au niveau transistor nous donnent une estimation du bruit ramené en entrée de $15\text{nV}/\sqrt{\text{Hz}}$, soit une DSP théorique égale à $156.5\text{dB}_v/\text{Hz}$.

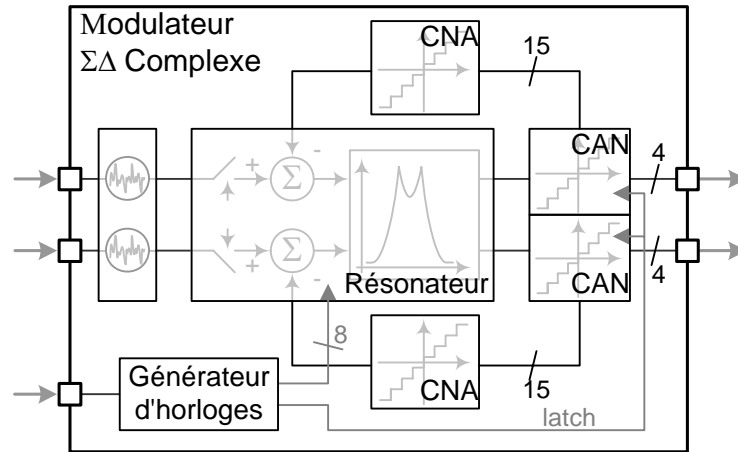


Figure 4 - 27 : Schéma bloc du Modulateur Sigma Delta Complexe

La structure étudiée étant particulièrement longue à simuler au niveau transistor (~ une journée de simulation pour obtenir quelques périodes en sortie), nous ne disposons d'aucune courbe nous permettant de faire une comparaison avec les caractéristiques fréquentielles de notre modèle.

IV.5.2. Simulations VHDL-AMS

Pour obtenir la courbe ci-dessous, nous effectuons une simulation temporelle suivie d'une analyse fréquentielle : un périodogramme des voies I et Q sur $N = 1048576$ échantillons avec des fenêtres de $NN = 131072$ entrelacées à 50% avec un fenêtrage Blackman7 ($F_{ech} = 13MHz$) :

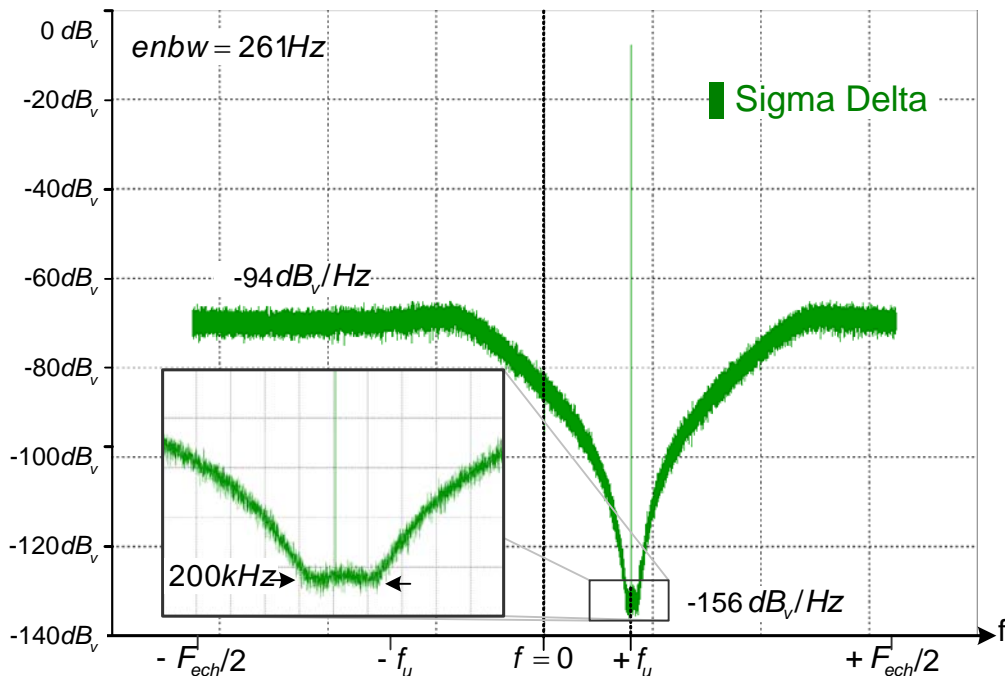


Figure 4 - 28 : Spectre du modulateur Sigma Delta complexe

Ce résultat est conforme aux mesures effectuées sur la puce. Nous pouvons maintenant nous intéresser à une simulation globale de la chaîne de réception Radio-Fréquence.

V. Modélisation de la chaîne de réception Radio-Fréquence

Comme énoncé dans l'introduction générale, les contre performances de la chaîne de réception Radio-Fréquence mesurées sur la puce RX n'ont pu être anticipées par la simulation, c'est ce qui a motivé ce travail de thèse. Nous avons développé au cours des chapitres et parties précédents un ensemble de théories et de modèles en VHDL-AMS prenant en compte diverses sources de bruit et destiné à expliquer les mesures. Il est maintenant temps de les utiliser.

V.1. La mesure du SNR

La principale mesure qui permet d'évaluer les performances de notre chaîne de réception est le rapport signal sur bruit (Signal to Noise Ratio). Ce terme, mesuré en sortie, désigne le rapport entre la puissance du signal utile et la puissance du bruit sommé (intégré) sur une bande de fréquence qui est généralement spécifiée, dans notre cas $\Delta f = 8\text{kHz}$. Pour cette mesure, la DSP de bruit (nécessaire au calcul du SNR) est obtenue par la moyenne, sur la bande utile $[f_u - 50\text{kHz}, f_u + 50\text{kHz}]$, des DSP issues du périodogramme d'un grand nombre d'échantillons répartis en sous-fenêtres de $N = 131072$ échantillons avec un fenêtrage de type Blackman7 ($enbw = 158.7\text{Hz}$).

Un exemple de mesure du SNR est donné par la figure ci-dessous :

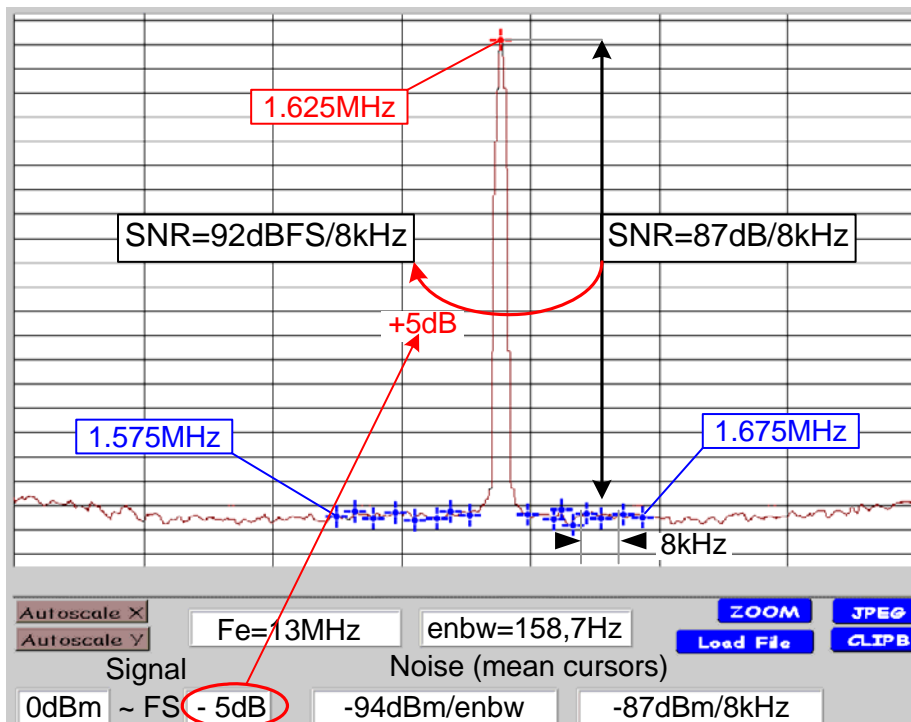


Figure 4 - 29 : Exemple de mesure du SNR (dBFS) sur banc

Les mesures du SNR s'effectuent à plusieurs amplitudes d'entrée, on définit alors le paramètre X :

$$X(\text{dB}) = 10 \cdot \text{Log}_{10} \left(\frac{P_{in}}{P_{FS}} \right) \quad \text{Eq. 4 - 7}$$

Où P_{in} désigne la puissance du signal d'entrée et P_{FS} la puissance du signal à pleine échelle (Full Scale). La pleine échelle correspond à l'amplitude maximum du signal d'entrée du Sigma Delta. Dans notre cas, elle correspond à un signal avec une puissance de -27.5dBm à l'entrée de la chaîne FI (avec un gain de 21dB pour l'amplificateur CAG) ou à un signal d'amplitude 800mV_{pp} sur chacune des voies d'entrée du modulateur. Soit une puissance en sortie du modulateur de 5dBm .

Une fois calculé, le SNR est ensuite normalisé par rapport à la variable X . Nous donnons ci-dessous la formule de calcul du SNR en dBFS :

$$\text{SNR}_{\text{dBFS}/8\text{kHz}}(FS - X) = \text{SNR}_{\text{dB}/8\text{kHz}}(FS - X) + X_{\text{dB}} \quad \text{Eq. 4 - 8}$$

Dans ces conditions, la courbe de mesure du SNR en dBFS est la suivante :

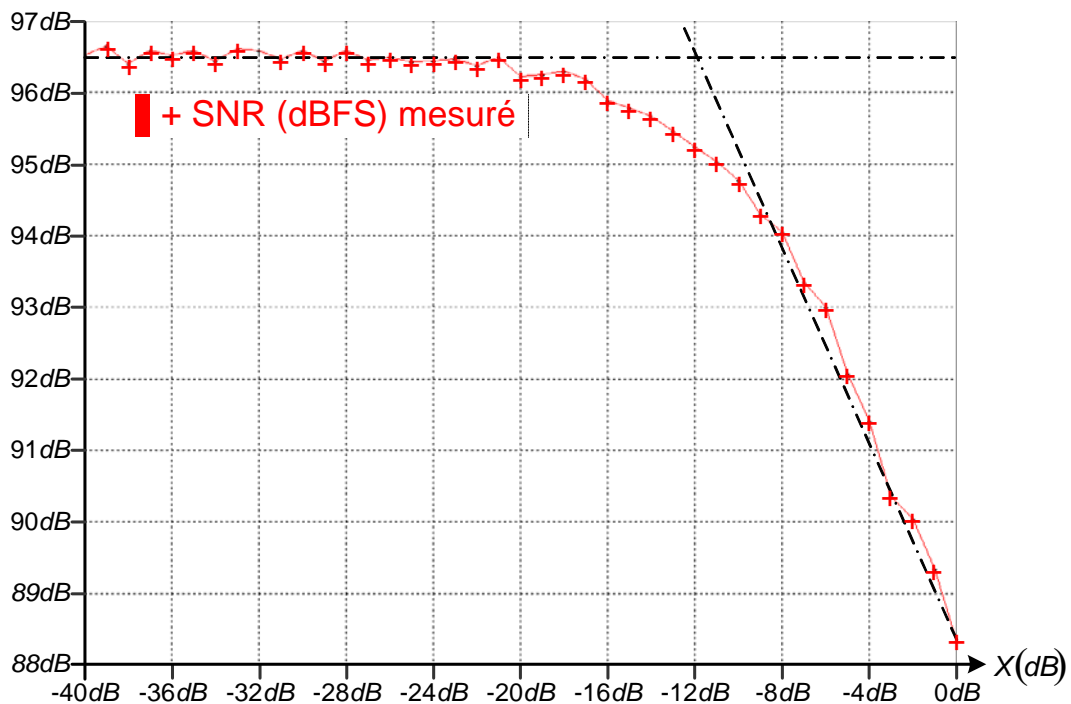


Figure 4 - 30 : SNR mesuré de la chaîne de réception

Comme effectué ci-dessus, nous pouvons séparer le graphique en deux zones distinctes :

La première, à faible amplitude, comprise entre $FS - 40\text{dB}$ et $FS - 20\text{dB}$, est plane et correspond au bruit thermique. Cette zone du graphique peut s'expliquer par le calcul suivant (cf. Eq. 4 - 5) :

$$\text{DSP}_{\text{thermique chaîne}} = 20 \cdot \text{Log}_{10} \left(\sqrt{ \left(\underbrace{10^{\frac{-143.7\text{dB}_v/\text{Hz}}{20}}}_{FI} \right)^2 + \left(\underbrace{10^{\frac{-156.5\text{dB}_v/\text{Hz}}{20}}}_{\Sigma\Delta} \right)^2 } \right) \approx -143.5\text{dB}_v/\text{Hz} \quad \text{Eq. 4 - 9}$$

Sur cette zone le $SNR_{8kHz}(dBFS)$ s'exprime alors de la manière suivante :

$$SNR_{FS-XdB/8kHz}(dBFS) = (20 \cdot \text{Log}_{10}(400e^{-3}) - X) - DSP_{thermique} - 10 \cdot \text{Log}_{10}(8e^3) + X \approx 96.5dB \quad \text{Eq. 4 - 10}$$

La seconde zone à forte amplitude est comprise entre $FS - 20dB$ et FS . On observe sur cette partie une importante altération ($\sim 8dB$) des performances à fort signal. Cette dégradation du SNR, due à un bruit proportionnelle à l'amplitude du signal, est typique des phénomènes de jitter que nous avons étudiés précédemment mais également des phénomènes liés au bruit de phase des générateurs. Nous allons donc nous attacher dans la partie suivante à déterminer les contributions respectives des différentes sources de bruit.

V.2. Les contributions en bruit

Comme nous l'avons dit en remarque, la mesure du SNR pour une amplitude donnée s'effectue à partir d'un grand nombre d'échantillons d'où un temps de mesure long. Afin de reproduire en simulation les conditions de mesure, nous réalisons un périodogramme des voies I et Q sur 15 fenêtres, entrelacées à 50%, de $NN = 131072$ échantillons chacune, soit $N = 1048576$ échantillons. La fréquence d'échantillonnage étant de $F_{ech} = 13MHz$, nous ne pouvons espérer avoir une résolution inférieure à $100Hz$. De plus, nous utilisons un fenêtrage (Blackman7) qui entraîne un étalement du spectre du signal. Afin de ne pas intégrer dans le calcul du SNR cet étalement du spectre du signal, nous excluons du calcul du SNR la bande de bruit $[f_u - 1kHz, f_u + 1kHz]$. Comme pour la mesure, nous utilisons l' Eq. 4 - 10 pour le calcul du SNR issu de la simulation.

V.2.1. Le jitter

La première source de bruit vers laquelle nous nous sommes tournés pour expliquer une partie de la dégradation du SNR mesuré est le jitter interne au modulateur Sigma Delta. D'après l'étude et les mesures réalisées dans le Chapitre III (cf. Eq. 3 - 19), le jitter vaut :

$$35ps \leq \sigma_{interne} \leq 45ps \quad \text{Eq. 4 - 11}$$

Nous rappelons l'expression de la Densité Spectrale de Puissance de l'échantillonnage jitté d'une sinusoïde (cf. Eq. 2 - 34) :

$$DSP_{ech\ jitter} = 10 \cdot \text{Log}_{10} \left((2 \cdot \pi \cdot f_u \cdot \sigma_{interne})^2 / F_{ech} \right) \quad \text{Eq. 4 - 12}$$

Ce qui équivaut à un bruit compris entre :

$$-140dB_c / Hz \leq DSP_{ech\ jitter} \leq -138dB_c / Hz \quad \text{Eq. 4 - 13}$$

En tenant compte de la contribution du bruit thermique :

$$SNR(FS) = \underbrace{-8dB_v}_{\text{Signal FS}} - 10 \cdot \text{Log}_{10} \left(\left(\underbrace{10^{\frac{-8dB_v + DSP_{\text{ech}} \text{ jitter} (dB_c / Hz)}{10}}}_{\text{Jitter}} + \underbrace{10^{\frac{-143.5dB_v / Hz}{10}}}_{\text{Thermique}} \right) \times 8kHz \right) \quad \text{Eq. 4 - 14}$$

Soit un SNR à pleine échelle :

$$95.1dBFS/8kHz \leq SNR(FS) \leq 94.6dBFS/8kHz \quad \text{Eq. 4 - 15}$$

Comme l'illustre la figure ci-dessous, nous sommes loin de 88.5dBFS/8kHz mesuré à pleine échelle. En fait, si la dégradation ne devait être imputée qu'au jitter interne, celui-ci correspondrait à $\sigma_{interne} \approx 140ps$.

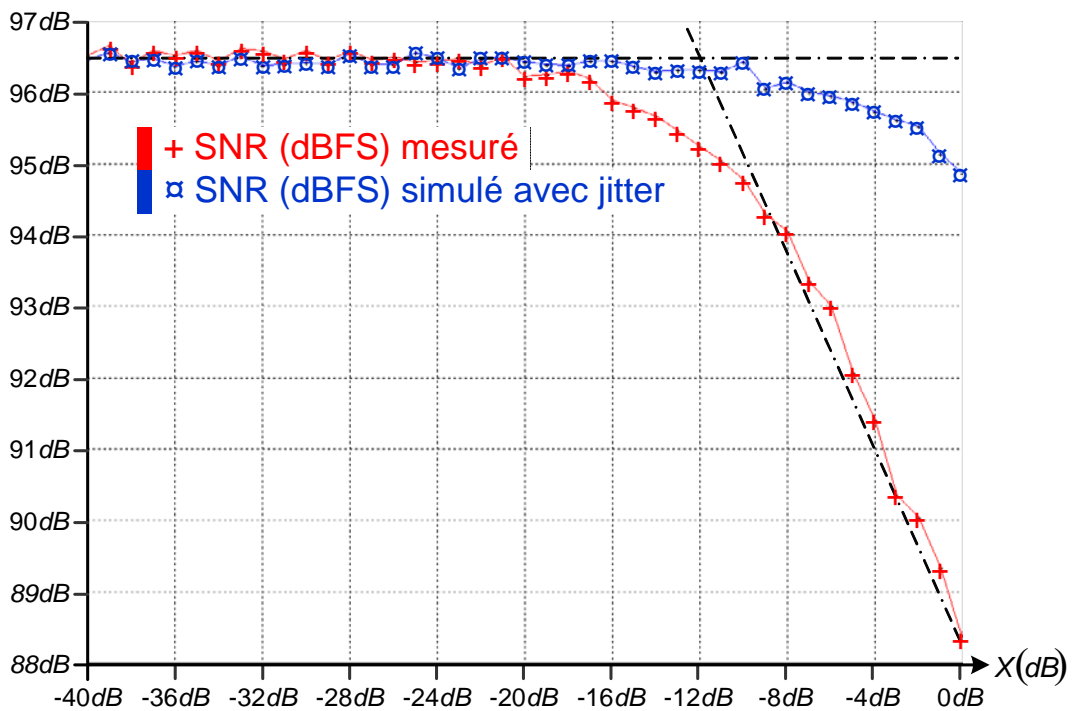


Figure 4 - 31 : SNR mesuré et simulé (avec un jitter de $\sigma_{interne} \approx 40ps$) de la chaîne de réception

Il nous faut donc soit reprendre les mesures du jitter interne effectuées dans le Chapitre III et reconsidérer les résultats, soit chercher ailleurs les raisons de la dégradation du SNR mesuré. Malgré les approximations que nous avons effectuées dans le Chapitre III, il nous semble peu vraisemblable d'avoir sous-estimé la valeur du jitter d'un facteur trois. Nous avons donc opté pour la seconde option.

V.2.2. Le bruit de phase

V.2.2.1. *Les sources*

La première source potentielle de bruit de phase a déjà été identifiée lors de l'étude de la chaîne FI c'est la PLL. Son bruit de phase a une DSP décrite par la Figure 4 - 17. Nous rappelons que le signal issu de la PLL a sa fréquence divisée par 4 lors de son passage dans le bloc de générateur de quadrature qui se comporte alors comme un générateur piloté en fréquence. Nous faisons l'hypothèse que le jitter interne du bloc de quadrature est négligeable face au bruit de son pilote (PLL). Nous supposons également que le bruit de la PLL est recopié sans effet cumulatif. Cette hypothèse a été vérifiée par une simulation temporelle suivie d'une analyse fréquentielle du modèle transistor du générateur de quadrature auquel nous avons appliqué le modèle bruité en VHDL-AMS de la PLL. De plus la fréquence étant 4 fois inférieure, la DSP du bruit de phase est plus faible de $12dB (= 10 \cdot \text{Log}_{10}(16))$. Cette dernière hypothèse a été vérifiée par des mesures explicitées dans la thèse de T. LAGUTERE. Ainsi dans la bande de bruit qui nous intéresse (entre $1kHz$ et $50kHz$ de la porteuse), nous trouvons une DSP de bruit de phase comprise entre $-125dB_c / Hz @ 1kHz$ et $-137dB_c / Hz @ 50kHz$. Nous rappelons que $DSP(f) = L(f) - 10 \cdot \text{Log}_{10}(2)$, la mesure étant ramenée en totalité sur les fréquences positives.

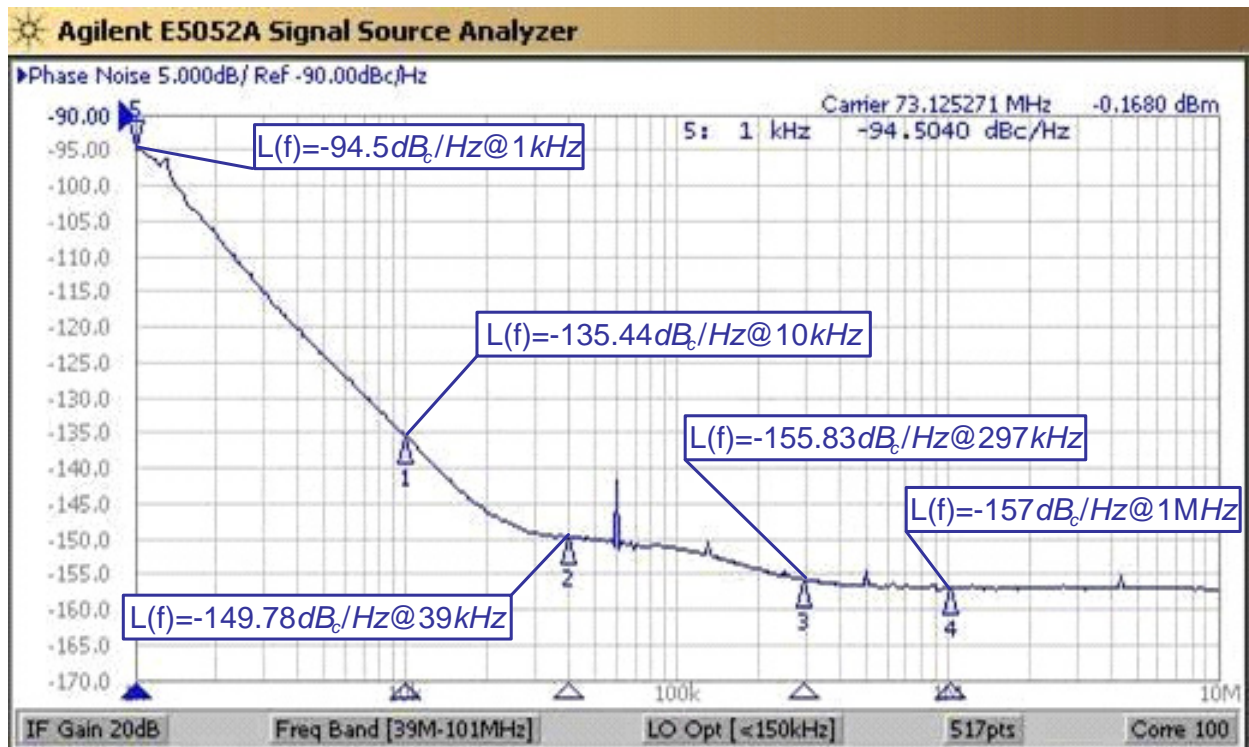


Figure 4 - 32 : Bruit du phase du générateur Rhode&Schwarz

La seconde source est une source externe. En effet, en consultant les spécifications du générateur utilisé pour délivrer la sinusoïde à l'entrée de la puce, nous constatons que son bruit de phase, même s'il peut paraître faible, est du même ordre de grandeur que le bruit de la PLL, du moins autour de $1kHz$. La DSP du bruit de phase du générateur, représentée ci-dessus (cf. Figure 4 - 32), a été mesuré chez EADS. Sa valeur semble conforme aux spécifications de la société Rhode&Schwarz, fabricant du générateur. La DSP de bruit du générateur dans la bande de calcul du SNR varie entre $-94.5dB_c/Hz@1kHz$ et $-151dB_c/Hz@50kHz$. En dépit de leur faible valeur, nous allons voir dans la partie suivante l'influence de ces deux sources de bruit sur le SNR de la chaîne de réception.

V.2.2.2. Le modèle théorique

Comme nous l'avons constaté à la fin de la partie III, la simulation du modèle en VHDL-AMS du bloc FI ne fonctionne pas correctement lorsque l'on ajoute le bruit de phase de la PLL. Or nous avons besoin d'estimer la contribution de ce bruit au SNR de la chaîne de réception RF. Nous nous sommes donc tournés vers une estimation de la propagation du bruit de phase dans cette structure. Considérons le schéma bloc suivant fondé sur les équations qui régissent le fonctionnement du mélangeur (Eq. 4 - 2 et Eq. 4 - 3) :

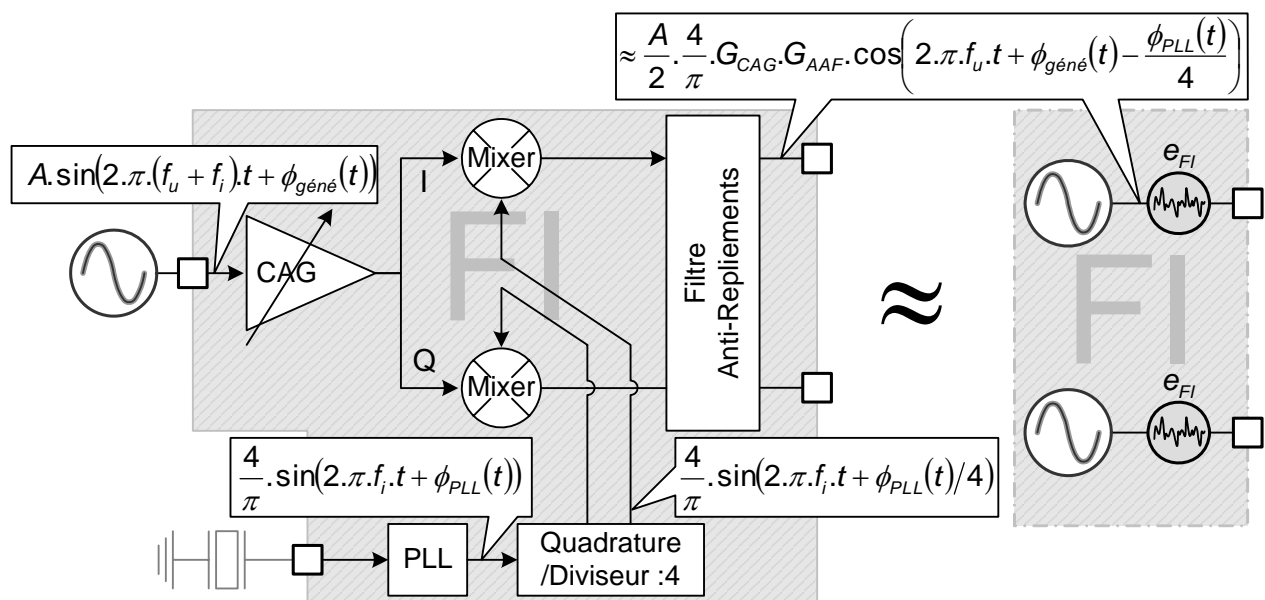


Figure 4 - 33 : Etude théorique de la propagation du bruit de phase dans le bloc FI

Du point de vue du bruit de phase, nous pouvons théoriquement remplacer l'ensemble du bloc FI par deux sources de tension en quadrature de phase ayant chacune un bruit de phase de

$\phi_{\text{généré}}(t) - \phi_{\text{PLL}}(t)/4$. Nous ajoutons également le bruit thermique équivalent $e_{F1} = \left(10^{\frac{-143.6 \text{ dB}_v / \text{Hz}}{20}} / \sqrt{2} \right)$
 $= 47.26 \text{ nV}_{\text{rms}} / \sqrt{\text{Hz}}$ (cf. Eq. 4 - 5).

Le modèle de bruit de phase du générateur $\phi_{\text{généré}}(t)$ utilise le principe de filtres cascades, précédemment utilisé (cf. Chapitre II), pour reproduire la forme du spectre en fréquence. Nous utilisons comme base une source de bruit blanc. Le modèle ainsi que son spectre sont disponibles en Annexe (cf. §IV.12)

Grâce à ce nouveau modèle simplifié du bloc FI, nous réalisons dans les mêmes conditions que précédemment de nouvelles séries de simulations afin d'estimer le SNR global. Afin de d'évaluer l'influence propre des bruits de phase, nous avons effectué deux série de simulations ($FS \rightarrow FS - 40 \text{ dB}$) : l'une avec seulement le bruit thermique et le bruit de phase, l'autre avec les trois sources type de bruit (thermique, de phase et jitter). Nous obtenons alors les courbes ci-dessous :

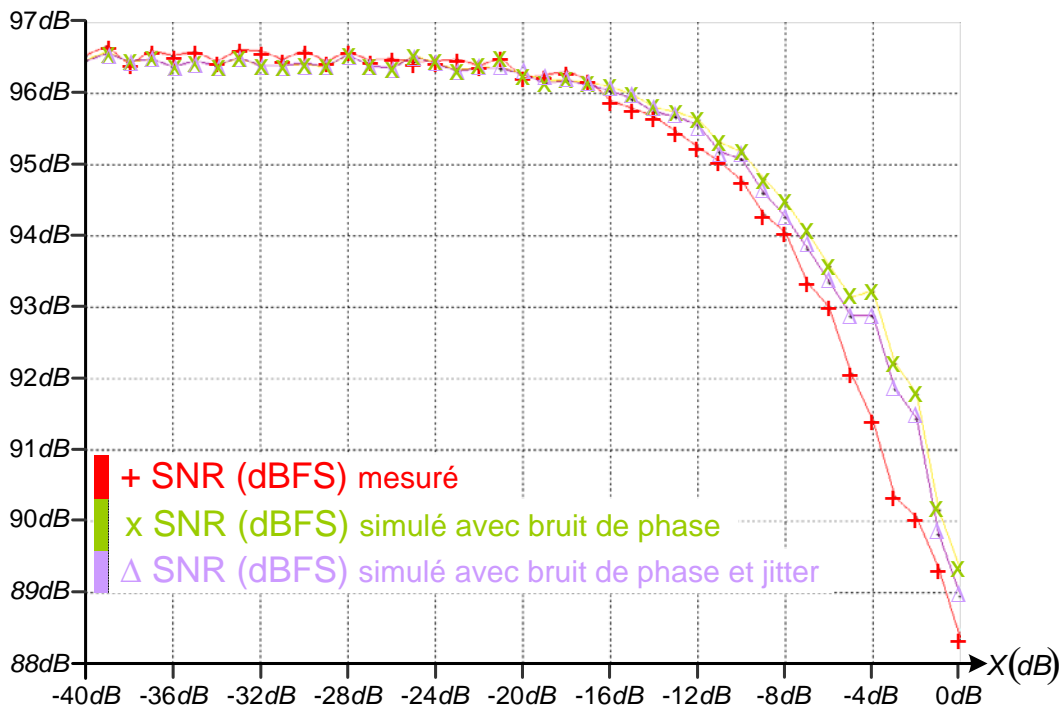


Figure 4 - 34 : SNR mesuré et simulé (avec bruit de phase et jitter) de la chaîne de réception

Même si les courbes de mesure et de simulation ne sont pas confondues, nous avons considérablement réduit l'écart entre elles. Comme il est impossible pour nous de reproduire à l'identique les conditions de mesures (ces mesures ont été réalisées par un service extérieur au notre), nous ne chercherons pas à obtenir une superposition des courbes de simulation et de mesure.

Si l'on compare les deux courbes issues de la simulation, on constate que les phénomènes de jitter n'entrent que faiblement dans la dégradation du jitter. De même en comparant la valeur des DSP des deux bruits de phase, on peut conclure que le bruit du générateur est prépondérant dans la dégradation du SNR et que les bruits internes n'apportent qu'une contribution minime. C'est un phénomène connu des spécialistes du test des convertisseurs analogique/numérique qui, bien souvent, au lieu de mesurer les performances de leur circuit de conversion, mesurent les performances du générateur utilisé pour le test.

VI. CONCLUSION

La finalité de ce dernier chapitre était d'établir la convergence des axes développés au cours des précédents chapitres, à savoir : l'étude et la modélisation des sources de bruit ainsi que la mesure du jitter d'une part et la modélisation d'une chaîne de réception RF d'autre part. Le but étant d'expliquer la dégradation du SNR et surtout de déterminer la part de chacune des sources de bruit responsables de cette dégradation. La chaîne de réception étant un circuit complexe, nous avons dans un premier temps étudié et modélisé la chaîne FI. La modélisation du modulateur Sigma Delta a ensuite suivi. Chaque bloc étant modélisé avec la source de bruit interne qui le caractérise (bruit thermique et jitter pour le générateur d'horloges non recouvrantes). Une fois l'ensemble des blocs modélisés, nous avons reproduit en simulation les conditions de mesure et simulé l'ensemble de la chaîne de réception RF. Nous avons pu alors déterminer les contributions individuelles de chaque source de bruit dans la dégradation du SNR.

CONCLUSION

Les travaux présentés dans ce mémoire sont une contribution à l'étude et à la modélisation comportementale des sources de bruit dans les circuits analogiques à temps discret. L'application de ces travaux est le bloc FI (Fréquence Intermédiaire) d'une chaîne de réception Radio Fréquence et plus particulièrement le modulateur Sigma Delta utilisé dans le processus de conversion analogique/numérique.

La tendance actuelle est l'intégration de fonctions de plus en plus complexes dans des ASIC mixtes pour des applications visant la radio logicielle. Les Radiocommunications Mobiles Professionnelles (PMR), qui nécessitent des performances supérieures à celle offertes par les solutions grand public type GSM, n'échappent pas à cette tendance. Comme le coût de développement et de fabrication d'un ASIC mixte reste élevé, il est primordial de pouvoir évaluer les performances d'un circuit avant son départ en fonderie.

Les outils de CAO précédemment spécialisés soit dans le domaine analogique soit dans le domaine numérique ont dû évoluer pour tendre vers un environnement de conception unifié. Etant donné la complexité des circuits mixtes précédemment cités, il s'avère impossible de faire des simulations bas niveau (au niveau des transistors pour la partie analogique et au niveau des bascules pour la partie numérique) de l'ensemble du système. D'où la nécessité de réaliser des modèles comportementaux plus haut niveau rendant possible des simulations mixtes globales. Le langage VHDL-AMS, que nous avons choisi pour nos travaux, permet une telle modélisation.

Afin d'appréhender correctement le domaine des chaînes de réception RF, nous avons débuté la présentation de nos travaux par un rappel des différents types d'architecture. Au regard des performances attendues par la PMR, nous avons pu justifier le choix d'une architecture comportant un bloc FI et une conversion analogique/numérique hors bande de base. Cette particularité a entraîné l'utilisation d'un modulateur Sigma Delta passe bande à temps discret pour la conversion des signaux analogiques. Ce type de modulation permet de rejeter le bruit de quantification hors de la bande utile. La particularité du modulateur employé tient à l'utilisation de cellules de HILBERT qui permettent de traiter simultanément les voies en quadrature de phase I et Q. Le spectre de sortie du modulateur est alors dissymétrique par rapport à l'axe des fréquences nulles. Le modulateur est alors dit « complexe » par opposition aux modulateurs « réels » qui ne traitent qu'une voie. Pour réaliser le filtre du convertisseur Sigma Delta des structures à capacités commutées ont été

employées. Cependant les techniques utilisées classiquement pour simuler ce type de structures (analyse transitoire ou fréquentielle), ne peuvent prendre en compte toutes les sources de bruit. Afin d'offrir aux concepteurs le moyen d'évaluer ces différents bruits, nous avons entrepris la création d'une bibliothèque de modèles comportementaux de sources de bruit. L'étude et la modélisation des différentes sources ont été développées dans le second chapitre. Ce chapitre débute par un inventaire des différentes sources de bruit. Cela nous a permis de différencier deux familles de bruit que nous modéliserons différemment : les bruits blancs (bruit thermique, bruit de grenaille, ...) et en $1/f$ (bruit de Flicker, ...) d'une part et les bruits cumulatifs en $1/f^2$ et $1/f^3$ (bruit de phase) d'autre part. Les outils de modélisation/simulation utilisés ne permettant pas une prise en compte directe du bruit dans le domaine fréquentiel, la seconde partie de ce chapitre s'intéresse aux relations mathématiques liant le domaine fréquentiel (Puissance, Densité Spectrale de Puissance, ...) et le domaine temporel (écart-type, fonction de transfert en z , ...). Chaque source de bruit a pu alors être modélisée dans le domaine temporel et un post traitement par FFT a permis de valider notre étude théorique. L'étape suivante a constitué à paramétrer les modèles à partir de mesures réalisées sur la puce ou bien de simulations au niveau transistor.

Une partie des paramètres physiques peut être estimée par des simulations RF (bruit thermique, propagation du bruit de phase, ...), l'autre partie est directement disponible dans les spécifications des appareils de mesures et la dernière partie n'est accessible que par la mesure (jitter interne). Le troisième chapitre décrit donc une méthode originale mise en œuvre pour estimer le jitter interne du modulateur Sigma Delta. Cette partie débute par une étude théorique qui justifie la démarche et qui fixe les domaines de validité de cette dernière. Une description de l'instrumentation et du banc de mesure suit. La dernière étape comporte l'interprétation des mesures et l'extraction du jitter interne, grâce aux équations définies dans la première partie de ce chapitre. A l'issue de ce chapitre nous disposons de suffisamment d'informations pour simuler correctement les sources de bruits internes et externes de la chaîne de réception RF.

Le dernier chapitre est consacré à la simulation globale du bloc comprenant la chaîne FI et le modulateur Sigma Delta. Bien que le VHDL-AMS soit compatible avec des simulations au niveau des transistors, nous avons opté pour une modélisation comportementale de chacun des blocs afin de réduire le temps de simulation. Les deux premières parties de ce chapitre sont consacrées à la description des modèles du bloc FI et du modulateur. Chaque modèle comportemental intègre une source de bruit interne (thermique, jitter, phase) et est comparé avec la modélisation au niveau des transistors. Par ailleurs, nous basant sur la mesure du SNR de la chaîne RF, nous avons entrepris d'expliquer a posteriori l'influence de chaque bruit sur les performances globales du circuit. Compte tenu des hypothèses retenues et des sources de bruit envisagées, nos simulations ont permis d'expliquer une grande partie de la dégradation du SNR.

L'objectif fixé par cette thèse était d'étudier et de modéliser les sources de bruit dans un système à temps discret. En ce sens, nous pouvons considérer que l'objectif est atteint. Le concepteur dispose en effet d'une bibliothèque qui permet de modéliser de nombreuses formes de bruit grâce au jeu de filtres en z. Pour chaque bloc de la chaîne RF, il dispose également de modèles de complexité variable suivant l'utilisation retenue.

En guise de perspectives à ces travaux, les points suivants sont à envisager :

- Il semble que les limites des outils de modélisation /simulation choisis aient été atteintes (Chapitre IV). Ainsi l'étude des effets du bruit de phase de la PLL sur le bloc FI, n'a pas été effectuée. En effet, on peut citer, parmi les difficultés imposées au simulateur, la présence de signaux de fréquences très distantes en ordre de grandeur et la gestion fine de la synchronisation entre noyaux numérique et analogique associée aux variations aléatoires, en amplitude et en temps, dues aux sources de bruit. De nombreux problèmes de convergence ont été rencontrés. Les simulateurs « système » (type ADS de AGILENT) seraient à priori mieux adaptés car pourvus algorithmes de simulation différents, capables de gérer la complexité de la chaîne étudiée. Dans le cadre d'un flot de conception complet et fluide, il faudrait donc assurer la conversion des modèles de bruit développés dans cette thèse, afin de reporter leurs caractéristiques dans de tels simulateurs.
- La seconde perspective touche à la réutilisation de nos travaux sur le bruit. En effet au regard du temps qui leur est imparti, il est primordial pour les concepteurs de pouvoir utiliser les modèles de bruit sans avoir à entrer dans le code VHDL-AMS. C'est dans cette optique qu'il faudrait mettre en place une interface permettant de générer automatiquement le modèle correspondant au bruit décrit par l'utilisateur. Cette action s'inscrirait dans une démarche plus globale de génération automatique de modèles.

BIBLIOGRAPHIES

I. BIBLIOGRAPHIE : Chapitre I

- [Abi95] A. A. ABIDI, "Direct-conversion Radio Transceivers for Digital Communications", IEEE J. Solid-State Circuits, Vol. 30, n°12, pp 1399-1410, 1995.
- [ASS95] Pervez M. AZIZ, Henrik V. SORENSEN, Jan avm der SPIEGEL, " Performance of complex noise transfer functions in bandpass and multi band Sigma Delta systems ", ISCAS '95., IEEE International Symposium on , vol. 1 , pages 641 -644 1995 .
- [Auv00] J. AUVRAY, " Systèmes électroniques 1999-2000", Modulation Δ , Université Pierre et Marie Curie IST SETI.
- [Bas00] A. BASCHIROTTI, " Switched capacitor filters ", The VLSI Handbook, ed. Wai-Kai Chen Boca Raton : CRC Press LLC, 2000.
- [BBL96-1] F. BAILLIEU, Y. BLANCHARD, P. LOUMEAU, H. PETIT, J. PORTE, " Capacités commutées et applications ", Chapitre 1 pages 4-8, DUNOD, 1996
- [BBL96-2] F. BAILLIEU, Y. BLANCHARD, P. LOUMEAU, H. PETIT, J. PORTE, " Capacités commutées et applications ", Chapitre 7 pages 135-142, DUNOD, 1996
- [Ben48] W. R. BENNETT, " Spectra of quantized signals ", bell systems technical journal, vol. 27, pages 446-472, july 1948.
- [Bre17] Brevet n°493660
- [Bil88] P. BILDSTEIN, " Filtres à capacités commutées ", pages 1-5, Techniques de l'ingénieur 12-1988.
- [Cou96] F. de COULON, " Théorie et traitement des signaux ", Chapitre premier pages 23-53, DUNOD, 1996.
- [Ded94] Ian DEDIC, " A sixth order triple loop Sigma Delta CMOS ADC with 90dB of SNR and 100kHz bandwidth ", digest of technical papers, international solid state circuits conference, pages 188-189, February 1994.
- [DeQ96] D. DECLERQ, A. QUINQUIS, « Le signal déterministe », collection : signaux et systèmes en question, Ed. Hermes, 1996.
- [Ger96] Deron L. GEROW, " Error mechanisms in Sigma Delta analog to digital converters ", Thesis, University of Maine, pages 1-4, august 1996.
- [Geg02] « software radio » ou « radio logicielle » www.tdf.fr/filemanager/download/138/
- [Hap66] Numéro spécial du Haut-parleur de mai 1966
- [Har28] R. HARTLEY "Single-Sideband Modulator," U.S Patent 1666206, April 1928
- [IYM62] H. INOSE, Y. YASUDA, J. MARAKAMI, " A telemetering system by code modulation, Delta Sigma modulation ", IRE Trans. On Space Electronics and Telemetry, Set-8, pages 204-209, September 1962.

- [JMS94] S. JANTZI, K. MARTIN, M. SNELGROVE, A. SEDRA, "A complex Bandpass $\Sigma\Delta$ converter for digital radio", proc, ISCAS, pages 453-456, 1994.
- [JSS91] S. JANTZI, R. SCHREIER, M. SNELGROVE, "Bandpass Sigma Delta analog to digital conversion", IEEE transactions on circuits and systems, pages 1406-1409, November 1991.
- [Lar94] Inventeurs et Scientifiques : Dictionnaire de bibliographie – Larousse 1994
- [Lar95] Le dictionnaire des inventeurs et inventions – Larousse 1995
- [Lee98] Thomas H. LEE, "The design of CMOS radio frequency integrated circuits", Chapter 18 pages 550-570, Cambridge University Press, 1998.
- [LeS87] W. L. LEE, C. G. SODINI, "A topology for higher order interpolative coders", Proceeding of the 1987 IEEE International Symposium on circuits and systems, vol. 4, pages 459-462, may 1987.
- [Max73] James Clerk MAXWELL, "A treatise on electricity and magnetism" Volume 2 page 420, Dover Publications Inc, 1873.
- [Nyq28] H. Nyquist, "Certain topic in telegraph transmission theory," AIEE Trans., page 617-644, 1928
- [NST97] S. R. NORSWORTHY, R. SCHREIER and G. C. TEMES, Delta Sigma Data Converters – Theory, Design and Simulation, Ed. IEEE Press, 1997
- [Raz97] B. RAZAVI, "Design Consideration for Direct-Conversion Receivers", IEEE Trans. Circuits and Syst. Part II, Vol. 44, pp1428-435, 1997
- [Rib91] D. B. RIBNER, "A comparison of modulator networks for high order oversampled $\Sigma\Delta$ Analog-to-Digital Converters" IEEE Transaction on circuits and systems, vol. 38, n°2 Feb. 1991
- [Rud97] J. C. RUDELL "A 1.9-GHz Wide-Band IF Double Conversion CMOS Receiver for Cordless telephone Applications," IEEE J. solid-State Circuits Vol.12 pp. 2071-2088 Dec. 1997
- [Son00] B. Song "Nyquist-Rate ADC and DAC" The VLSI Handbook, Ed. Wai-Kai Chan, Boca Raton : CRC Press LLC, 2000.
- [SrS77] A.B. SRIPAD and D.L. SNYDE, "A necessary and sufficient condition for quantization errors to be uniform and white", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASS P-25, pp 442-448, Oct.1977.
- [Ste75] R. STEELE, "Delta modulation systems", Pentech Press, London, 1975.
- [Par90-1] / S. PARK, "Principles of Sigma Delta modulation for analog to digital converters", section [Par90-2] 2-3 [Par90-1] / section 5 [Par90-2], Motorola, APR8/D Rev.1, 1990.
- [Wea56] D.K. WEAVER "A Third Method of Generation and Detection of Single-Sideband Signals," Proc. of IRE, Vol. 44, 1956, pp1703-1705.

II. BIBLIOGRAPHIE : Chapitre II

- [ADV00] ADVanceMS reference manual, Mentor Graphics, 00.
- [BaA66] J. A. BARNES and D. W. ALLAN, A Statistical Model of Flicker noise, Proceedings of the IEEE, vol. 54, N° 2, pp. 176-178, Feb 66.
- [BoM58] BOX, G.E.P, M.E. MULLER, A Note on the Generation of Random Normal Deviates, Annals Math, 58.
- [BoW88] B.E.BOSER and B.A.WOOLEY, The Design of Sigma-Delta Modulation A-to-D Converters, IEEE journal of solid-state circuits, vol. 23, N° 6, pp. 1298-1308 Dec 88
- [CuS66] L.S. CUTLER, C.L. SEARLE, Some Aspects of the Theory and Measurement of Frequency Fluctuations in Frequency Standards, Proc. of the IEEE, vol. 54, N° 2, pp. 136-154, Feb 66
- [DeQ96] D. DECLERCQ A. QUINQUIS, Le signal aléatoire, ed. Hermes 96.
- [EDA] <http://www.eda.org/verilog-ams>
- [Eld04] Eldo RF User's Manual, Software Version 6.3_1, Release 2004.1 Mentor Graphics
- [HaL99] A. HAJIMIRI and T.H. LEE , The Design of Low Noise Oscillators, 1999. ed. KAP
- [Har78] F.J.HARRIS, On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, Proceedings of the IEEE vol. 66, N°. 1,pp. 51-83 Jan 78
- [Her02] Y. Hervé, VHDL-AMS : applications et enjeux industriels, Dunod, Paris, 2002.
- [HeR99] F. HERZEL and B. RAZAVI, A study of Oscillator jitter Due to Supply and Substrate Noise, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 46, N° 1, Jan 99
- [IEE99] IEEE Standard VHDL Analog and Mixed-Signal Extensions, IEEE Std 1076.1 – 1999, IEEE, New-York, 1999.
- [Joh28] JOHNSON : "Thermal Agitation of Electricity in Conductors," Phys. Rev., v. 32, July 1928, pp 97-109
- [Kun98] K. KUNDERT, Modelling and simulation of jitter in PLL Frequency synthesisers, Cadence confidential manuscript, Dec 98.
- [Lee98] T.H. LEE "The Design of CMOS Radio-Frequency Integrated Circuits" Cambridge University Press, 1998
- [LML04] G. MONNERIE, H. LEVI, N. LEWIS, P. FOUILLAT, "Behavioral modeling of noise in discrete time systems with VHDL-AMS - Application to a Sigma-Delta modulator ", IEEE ICIT 2004 International Conference on Industrial Technology, Hammamet, Tunisie, 8-10 Décembre 2004

- [LMS04] N. LEWIS, G. MONNERIE, J. SABATIER, P. MELCHIOR, M. ROBBE, H. LEVI Using fractional differentiation for the modeling of $1/f$ noise – Application to discrete-time noise sources in VHDL-AMS, 2004 IEEE International Symposium on Industrial Electronics, ISIE'04 4-7 Mai 2004, Ajaccio - France
- [MLD03] G. MONNERIE, N. LEWIS, D. DALLEY, H. LEVI, M. ROBBE, Modelling of transient noise sources with VHDL-AMS and normative spectral interpretation, FDL'03, Forum on Design Languages, Sept. 23-26, 2003, Frankfurt, Germany
- [MPR99] Top down design of high performance Sigma Delta modulators, F MEDEIRO, B. PEREZ-VERDU and A. RODRIGUEZ ed. Kluwer Academic Publisher 1999.
- [Nei01] J. Mc NEILL, "A simple method for relating time and frequency domain measures of oscillator performance" , Proc. IEEE 2001 SSSMSD2001, Austin, Texas, Feb 2001.
- [Nut81] A.H.NUTTALL, Some Windows with Very Good Side lobe Behavior, IEEE Transactions on acoustics, speech, and signal processing, vol. ASSP-29 N°. 1, pp. 84-91 Feb 81
- [Nyq28] NYQUIST : "Thermal Agitation of Electric Charge in Conductors," Phys. Rev., v. 32, July 1928, pp 110-113
- [Ous83] A. OUSTALOUP – Systèmes Asservis Linéaires d'Ordre Fractionnaire – Editions Masson, 1983
- [Ous95] A. OUSTALOUP – La dérivation non entière : théorie, synthèse et applications dans les sciences pour l'ingénieur- Editions Hermès, 1995
- [SCI] scilab-2.7 SCILAB GROUP (INRIA, ENPC) www-rocq.inria.fr/scilab/
- [Sch18] SCHOTTKY : "über spontane Stromschwankungen in verschiedenen Elektrizitätsleitern" Annalen des Physik, v. 57, 1918, pp. 541-567
- [SMA] SMASH documentation 4.9.12, Dolphin Integration
- [StM97] P. STOICA and R. MOSES, Introduction to Spectral Analysis, Prentice-Hall, Englewood Cliffs, NJ, 97
- [VER] <http://www.verilog-ams.com>
- [WaH04] A.P.WAGNER, J. HAASE, Monte Carlo Simulation Using VHDL-AMS, FDL04.
- [Wel67] P.D.WELCH, The Use of Fast Fourier Transform for Estimation of Power Spectra : A Method Based on Time Averaging Over Short, Modified Periodogram, IEEE Transaction on Audio and Electro-acoustics, vol. AU-15, N°. 2, pp. 70-73 June 67
- [Zie62] A. van der ZIEL : "Thermal noise in Field effect transistors," Proc. IEEE, August 1962, pp 1801 1812.
- [Zie86] A. van der ZIEL : "Noise in solid state devices and circuits," Wiley New York 1986.

III. BIBLIOGRAPHIE : Chapitre III

- [ADV00] ADVanceMS reference manual, Mentor Graphics, 00.
- [AD9240] http://www.analog.com/UploadedFiles/Data_Sheets/417954540AD9240_a.pdf
- [AMS] <http://asic.austriamicrosystems.com/>
- [BCC71] J.A. BARNES, A.R. CHI, L.S. CUTLER, D.J HEALEY, D.B. LEESON, T.E. MCGUNIGAL, J.A. MULLEN, W.L. SWITH, R.L. SYDNOR, R.F.C. VESSOT and G.M.R. WINKLER, Characterization of frequency stability, IEEE Transactions on Instrumentation and Measurement, vol. 20, pp. 105-120, 1971.
- [DMB94] D. DALLET, P. MARCHEGAY and M. BENKAIS, Jitter measurement of an ADC by statistical analysis, INT. J. Electronics, vol. 77, N°4, pp. 517-523, March 94.
- [Eld04] Eldo RF User's Manual, Software Version 6.3_1, Release 2004.1 Mentor Graphics
- [IEE99] IEEE Standard VHDL Analog and Mixed-Signal Extensions, IEEE Std 1076.1 – 1999, IEEE, New-York, 1999.
- [SAW90] M.SHINAGAWA, Y. AKAZAWA and T. WAKIMOTO, Jitter analysis of high-speed sampling systems, journal of solid-state circuits, vol. 25, N°. 1, pp. 220-224, Feb. 90.
- [SCI] scilab-2.7 SCILAB GROUP (INRIA, ENPC) www-rocq.inria.fr/scilab/

ANNEXES

SOMMAIRE

I. ANNEXES : CHAPITRE I	184
I.1. Annexe 1 : L'échantillonneur	184
I.1.1. L'échantillonnage	184
I.1.2. L'échantillonneur réel	185
I.2. Annexe 2 : Des filtres réels aux filtres complexes	187
I.2.1. Les filtres réels	187
I.2.2. Les filtres complexes	188
I.2.3. Exemple de transposition de filtres réel en filtres complexes	190
I.3. Annexe 3 : Les imperfections des interrupteurs et des capacités MOS	191
I.3.1. La non-linéarité de la résistance de conduction	191
I.3.2. Technique de linéarisation de la résistance de conduction	192
I.3.3. Les phénomènes parasites d'injection d'horloge	192
I.3.3.1. Les capacités parasites des interrupteurs MOS	192
I.3.3.2. Le phénomène d'injection d'horloge	193
I.3.4. Techniques de compensation des phénomènes parasites	194
I.3.4.1. Méthode utilisant les propriétés des PMOS	194
I.3.4.2. Méthode utilisant les chronogrammes	195
I.3.5. Les capacités parasites des capacités en poly-silicium	197
II. ANNEXES : CHAPITRE II	198
II.1. Annexe 1 : Les sources de bruit composées	198
II.1.1. L'addition d'un bruit blanc et d'un bruit de Flicker	198
II.1.2. L'addition d'un bruit en $1/f^2$ et d'un bruit en $1/f^3$	199
II.2. Annexe 2 : Modulation d'une source de bruit	200
II.2.1. Modulation d'une sinusoïde	200
II.2.2. Modulation d'une somme finie de sinusoïdes	201
II.2.3. Modulation d'une somme infinie de sinusoïde	202
II.2.4. Conclusion	203
II.3. Annexe 3 : Modulation de sources composées	204
II.3.1. Modulation d'une source de bruit issue de l'addition d'un bruit blanc et d'un bruit de Flicker	204
II.3.2. Modulation d'une source de bruit issue de l'addition d'un bruit en $1/f^2$ et d'un bruit en $1/f^3$	205

II.4. Annexe 4 : Synthèse d'un filtre d'ordre non-entier	206
II.4.1. Brique élémentaire	206
II.4.2. Filtre complet	208
II.5. Annexe 5 : Courbes sur l'intervalle $[-F_{ech}/2, F_{ech}/2]$	211
II.5.1. Bruit de Flicker et bruit blanc	211
II.5.2. Bruit de phase	211
II.5.3. Sinusoïde bruitée	212
II.5.4. Echantillonnage bruité	212
III. ANNEXES : CHAPITRE III	213
III.1. Annexe 1 : Layout du circuit « Gain_10 »	213
III.2. Annexe 2 : Approximation de l'écart type de $\alpha.A.2.\pi.F_{ech} \cdot \cos(\phi_e) \sum_{i=0}^{i=n-1} \beta^i \cdot \delta_{total}^i$	214
III.2.1. Insuffisance de l'approche théorique	214
III.2.2. Simulation d'un modèle idéal de filtre	217
III.2.3. Simulation d'un modèle MOS du filtre	218
IV. ANNEXES : CHAPITRE IV	220
IV.1. Modèle de l'amplificateur à Contrôle Automatique de Gain (CAG)	220
IV.2. Modèle du générateur de quadrature	221
IV.3. Modèle de Mélangeur	222
IV.4. Modèle du filtre Anti-repliement (Anti Aliasing Filter)	223
IV.5. Modèle de la PLL	225
IV.5.1. Fonction du bruit de phase de la PLL	225
IV.5.2. Modèle de la PLL	225
IV.6. Modèle de la chaîne FI	226
IV.7. Modèle du générateur d'horloges non-recouvrantes	227
IV.8. Modèle du Convertisseur Analogique/Numérique	228
IV.8.1. Le quantificateur	228
IV.8.2. L'encodeur logique	229
IV.9. Modèle du Convertisseur Numérique/Analogique	230

IV.10.	Modèle du résonateur	230
IV.11.	Modèle du modulateur Sigma Delta	232
IV.12.	Modèle de bruit de phase du générateur Rhode&Schwarz	233

I. ANNEXES : Chapitre I

I.1. Annexe 1 : L'échantillonneur

I.1.1. L'échantillonnage

Echantillonner un signal permet de passer d'une représentation temps continu $x(t)$ à une représentation discrète $x^*(n)$. Comme l'illustre la Figure 1 - 52 l'échantillonnage peut se représenter, mathématiquement, par la multiplication du signal par un peigne de Dirac dans le domaine temporel :

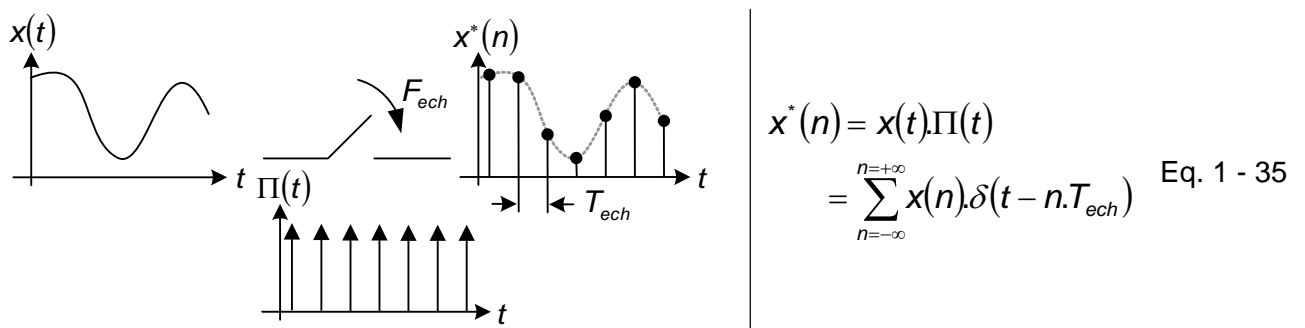


Figure 1 - 52 : Echantillonnage dans le domaine temporel

où T_{ech} désigne la période d'échantillonnage et $x(n) \equiv x(n \cdot T_{ech})$ les échantillons du signal $x(t)$ prélevés aux instants $n \cdot T_{ech}$.

Cette opération a une influence sur la représentation spectrale du signal, comme l'illustre la figure suivante sur un signal de largeur de bande égale à : $2 \cdot f_b$. Nous constatons en effet qu'autour de chaque multiple de la fréquence d'échantillonnage, s'effectue une recopie du spectre du signal échantillonné. Le spectre est donc périodique, de période F_{ech} .

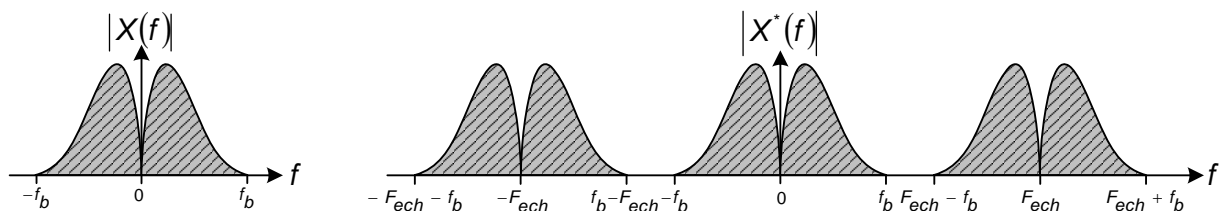


Figure 1 - 53 : Spectre d'un signal échantillonné

Le calcul de la Transformée de Fourier du signal échantillonné (cf. Eq. 1 - 36) corrobore cette constatation : le spectre du signal échantillonné est égal au produit de convolution « * » du spectre du signal et du peigne de Dirac en fréquence.

$$\begin{aligned}
 X^*(f) &= TF(x(t)) * TF(\Pi(t)) \\
 &= F_{ech} \cdot \sum_{n=-\infty}^{n=+\infty} X(f - n.F_{ech})
 \end{aligned}
 \tag{Eq. 1 - 36}$$

Suite à ce calcul, il paraît évident que l'échantillonnage soit soumis à certaines règles [Par90-1]. En effet afin de ne pas polluer l'information, il est préférable d'éviter tout recouvrement du spectre lors de la discrétisation du signal. Ce recouvrement a lieu lorsque la fréquence d'échantillonnage est mal appropriée ou que la bande utile est grande, comme le montre la figure suivante :

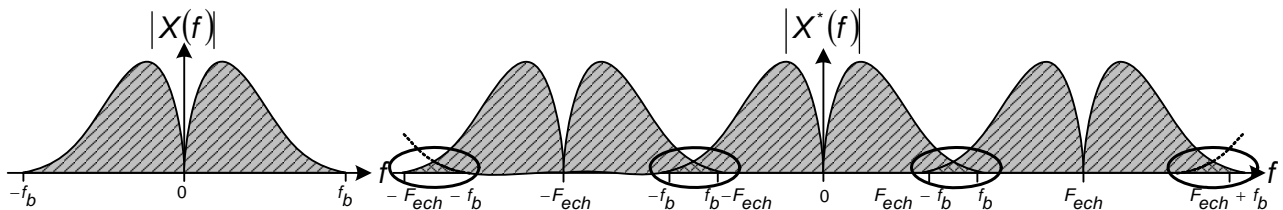


Figure 1 - 54 : Recouvrement de spectre

Graphiquement nous vérifions la condition de Shannon : la fréquence d'échantillonnage doit au moins être égale à deux fois la largeur du spectre du signal ($F_{ech} \geq 2.f_b$). $2.f_b$ est appelée fréquence de Nyquist [Nyg28].

Il est parfois difficile d'appliquer cette condition notamment lorsque le signal a une largeur de spectre infinie ou très grande. Dans ces conditions l'utilisation de filtres anti-repliement est préconisée. Ces filtres permettent de borner le spectre du signal avant l'échantillonnage et donc d'éviter tout repliement une fois le signal discrétisé :

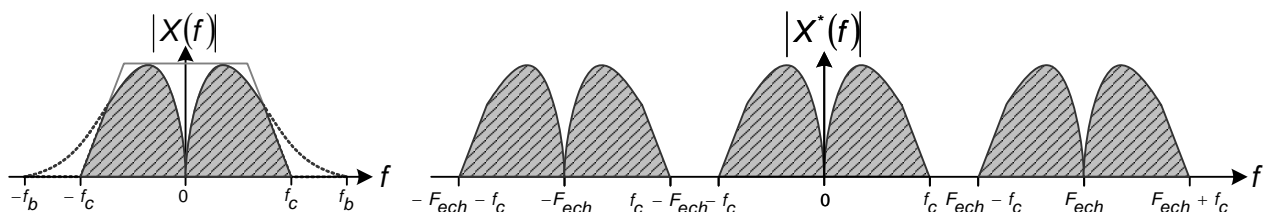


Figure 1 - 55 : Effet d'un filtre anti-repliement

1.1.2. L'échantillonneur réel

L'échantillonnage, comme nous venons de le décrire, n'est pas réalisable physiquement. En effet l'échantillonneur réel ne peut être basé sur l'utilisation de pulsations d'horloge de largeur infiniment petite (impulsion de Dirac) comme cela était fait.

Pour s'approcher des conditions réelles d'utilisation, nous considérons, comme décrite ci-dessous, une fonction impulsion $r(t)$ de largeur τ et de période T_{ech} :

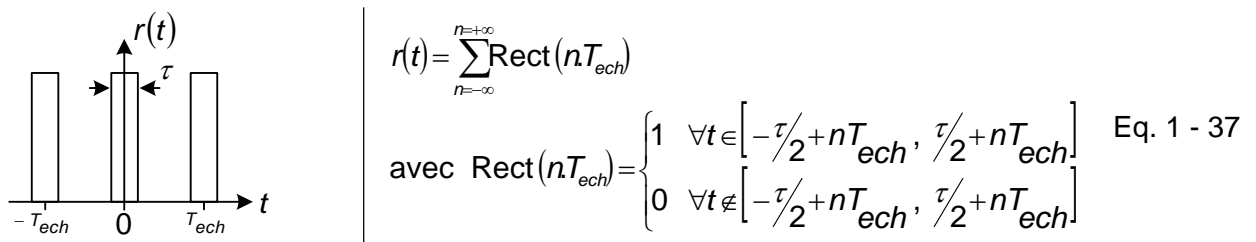


Figure 1 - 56 : Impulsion d'échantillonnage

La transformée de Fourier de $r(t)$ est :

$$TF(r(t)) = \tau \cdot F_{ech} \cdot \sum_{n=-\infty}^{+\infty} \frac{\sin(n \cdot \pi \cdot \tau \cdot F_{ech})}{n \cdot \pi \cdot \tau \cdot F_{ech}} \cdot \delta(f - n \cdot F_{ech})$$
 Eq. 1 - 38

Comme précédemment (cf. Eq. 1 - 36) la Transformée de Fourier du signal ainsi échantillonné est égale au produit de convolution des Transformées de Fourier des signaux $r(t)$ et $x(t)$:

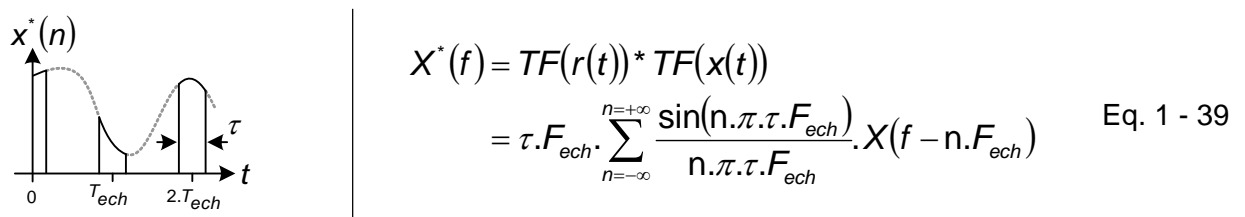


Figure 1 - 57 : Echantillonnage réel d'un signal $x(t)$

Suivant les valeurs de τ plusieurs représentations sont possible :

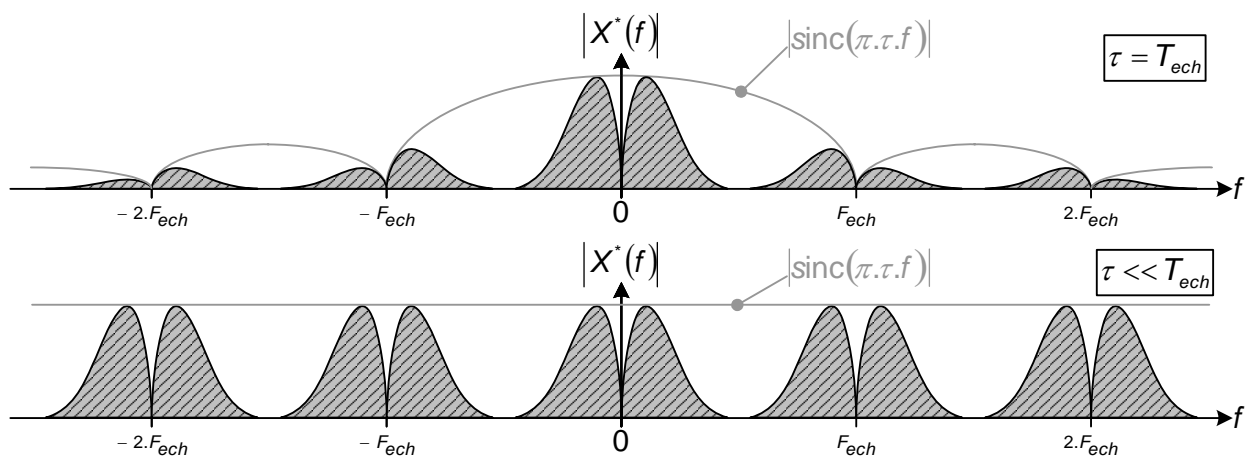


Figure 1 - 58 : Spectre d'un signal échantillonné

I.2. Annexe 2 : Des filtres réels aux filtres complexes

Nous nous intéressons, dans cette annexe, aux signaux entrants $A^*(f)$ et sortants $B^*(f)$ d'un filtre. Leurs composantes respectives $(A_I^*(z); A_Q^*(z))$ et $(B_I^*(z); B_Q^*(z))$ sont déphasées de $\pi/2$ l'une par rapport à l'autre :

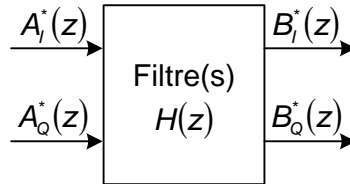


Figure 1 - 59 : Composantes entrantes et sortantes de filtre

Comme nous sommes dans des systèmes échantillonnés, nous utiliserons indifféremment la représentation sur le plan des fréquences ou sur le plan z :

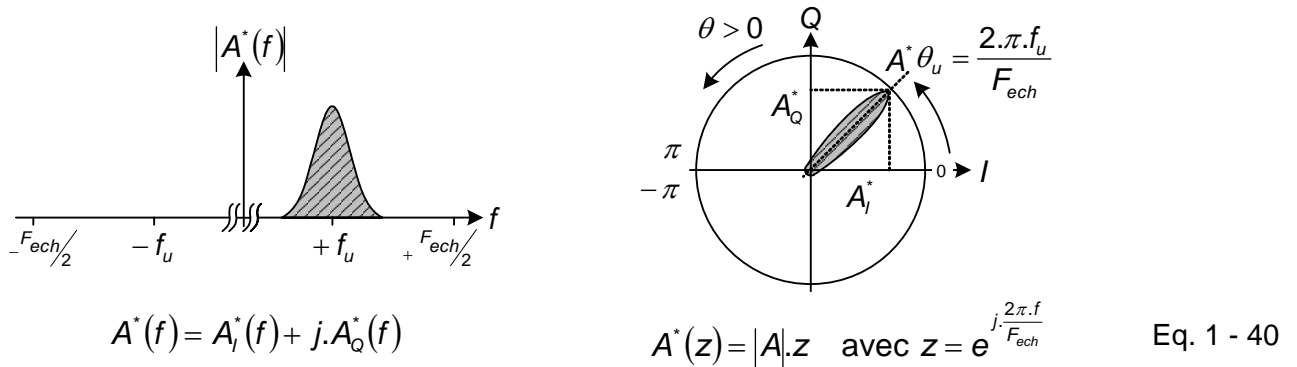


Figure 1 - 60 : Représentations dans le plan des fréquences et dans le plan en z

I.2.1. Les filtres réels

Nous utilisons, dans un premier temps, deux filtres réels $H_I(f)$ et $H_Q(f)$ (un par voie) pour le filtrage des composantes (ou voies) I et Q de $A^*(f)$. $B^*(f)$ s'exprime alors :

$$B^*(f) = \underbrace{H_I(f)A_I^*(f)}_{B_I^*(f)} + j \underbrace{H_Q(f)A_Q^*(f)}_{B_Q^*(f)} \quad \text{Eq. 1 - 41}$$

Nous supposons maintenant que les deux filtres sont identiques $H_I(f) = H_Q(f) = H(f)$. Comme les filtres sont monovoies, la représentation spectrale de leur gabarit est symétrique par rapport à l'axe des y dans une représentation spectrale classique et également symétriques par rapport à l'axe I dans le plan en z .

Un exemple de représentation, incluant une représentation sur le plan z, est donné ci-dessous :

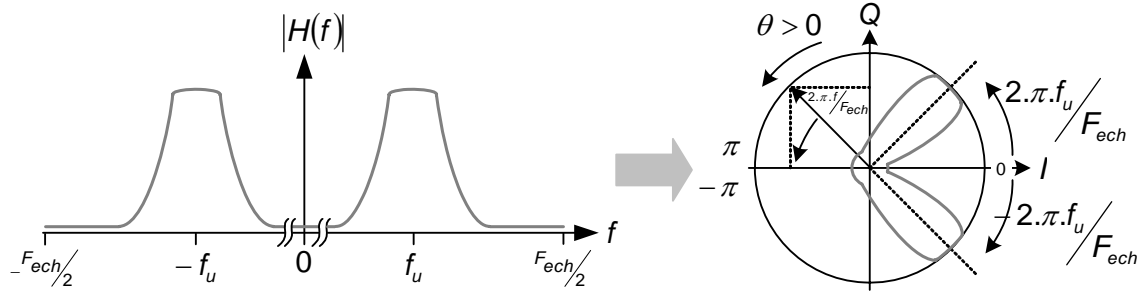


Figure 1 - 61 : Gabarit dans le domaine fréquentiel d'un filtre réel passe bande

Nous obtenons alors un filtre passe bande symétrique puisque le signal sortant du filtre s'exprime :

$$B^*(f) = |H(f)| \cdot e^{j\phi_H(f)} \cdot A^*(f) \quad \text{Eq. 1 - 42}$$

1.2.2. Les filtres complexes

Pour obtenir un filtre complexe nous utilisons conjointement les voies I et Q et leur déphasage. Considérons le filtre passe bas réel $H(f)$ appliqués respectivement à la voie I et à la voie Q :

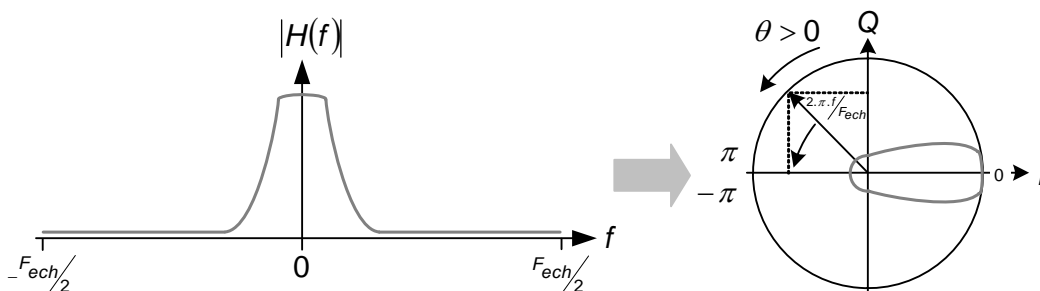


Figure 1 - 62 : Gabarit dans le domaine fréquentiel d'un filtre réel passe bas

Comme décrit ci-dessous, nous appliquons un vecteur de rotation $R(\theta_r) = \cos \theta_r - j \sin \theta_r$ [JMS94] [ASS95] à l'entrée du système, et effectuons par la même un changement de repère :

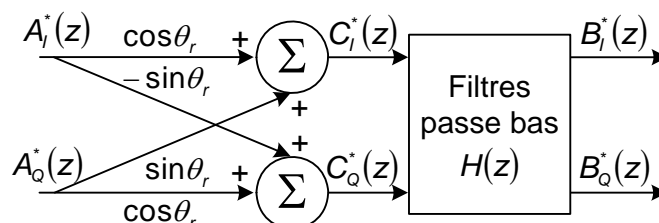


Figure 1 - 63 : Mise en place d'un vecteur rotation

Cette transformation est connue sous le nom de transformation d'Hilbert. Nous pouvons écrire dans le domaine temporel :

$$\begin{aligned}
 A(t) &= \underbrace{A \cos(2\pi f t)}_{A_I(t)} + j \underbrace{A \sin(2\pi f t)}_{A_Q(t)} \\
 \Rightarrow C(t) &= A((\cos \theta_r \cdot A_Q(t) + \sin \theta_r \cdot A_I(t)) + j(\cos \theta_r \cdot A_I(t) - \sin \theta_r \cdot A_Q(t))) \\
 &= A(\cos(2\pi f t - \theta_r) + j \sin(2\pi f t - \theta_r)) \\
 &= A(t) e^{-j \theta_r}
 \end{aligned}
 \tag{Eq. 1 - 43}$$

Nous pouvons alors écrire dans le domaine fréquentiel :

$$\begin{aligned}
 B^*(f) &= |H(f)| e^{j \phi_H(f)} \cdot C^*(f) \\
 &= |H(f)| e^{j(\phi_H(f) - \theta_r)} \cdot A^*(f)
 \end{aligned}
 \tag{Eq. 1 - 44}$$

Ce qui est équivalent à une rotation d'un angle $\theta_r = 2\pi f_r / F_{ech}$ du gabarit du filtre réel sur le plan z, ou d'une translation de f_r sur l'axe des fréquences comme cela est illustré par la Figure 1 - 64.

$$H(f) = \frac{H_o}{1 + j \frac{f}{f_b}} \xrightarrow{f' = f + f_r} H(f') = \frac{H_o}{1 + j \frac{f' - f_r}{f_b}}
 \tag{Eq. 1 - 45}$$

En prenant $f_r = f_u$ nous obtenons alors un filtre passe bande complexe (non symétrique) centré autour de notre bande utile :

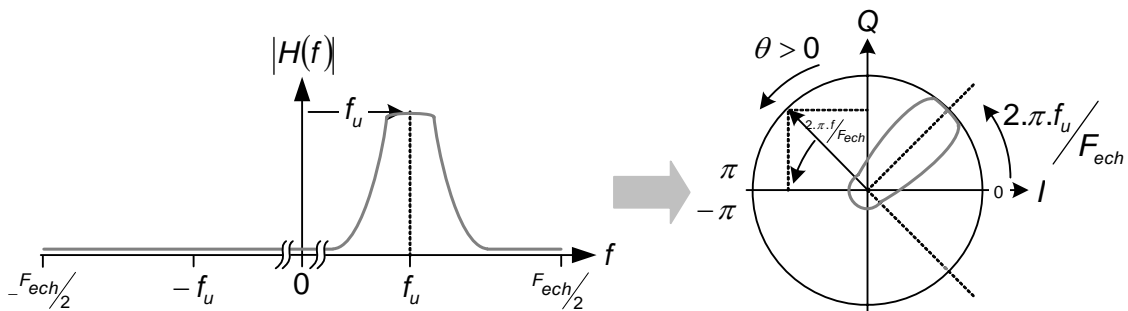


Figure 1 - 64 : Gabarit dans le domaine fréquentiel d'un filtre passe bande complexe

I.2.3. Exemple de transposition de filtres réel en filtres complexes

La multiplication par le vecteur rotation a bien entendu des conséquences quant à la construction des filtres en z . Nous donnons ci-dessous deux exemples simples de réalisation :

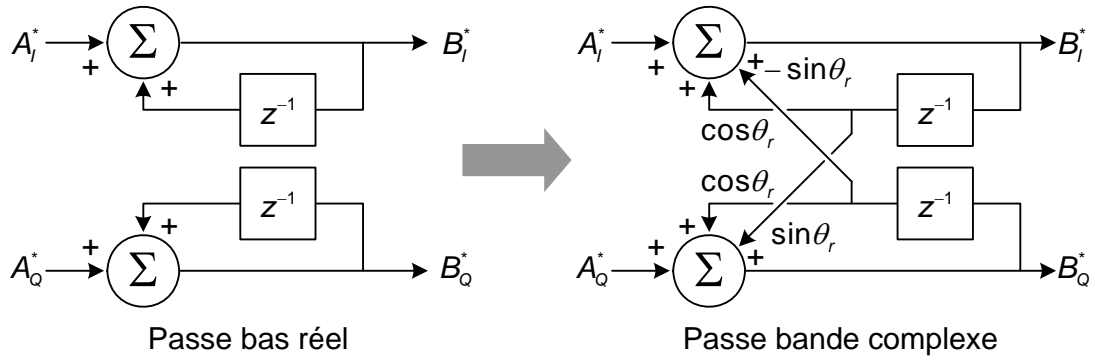


Figure 1 - 65 : Transposition complexe du filtre $H(z) = \frac{1}{1-z^{-1}}$

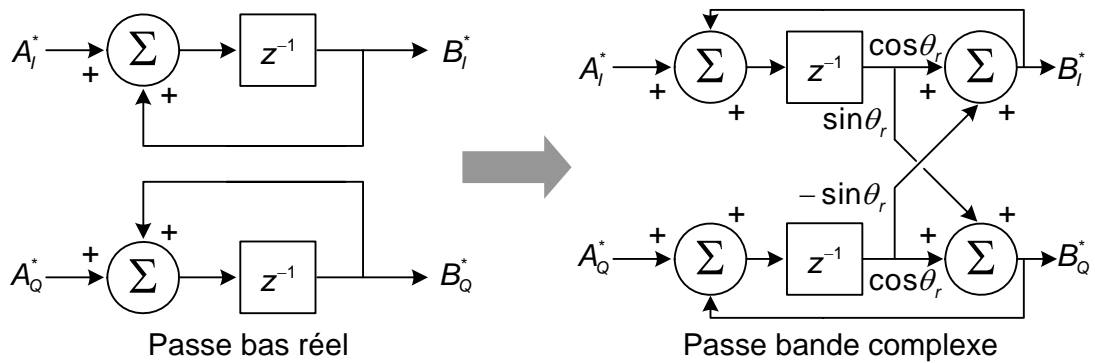


Figure 1 - 66 : Transposition complexe du filtre $H(z) = \frac{z^{-1}}{1-z^{-1}}$

I.3. Annexe 3 : Les imperfections des interrupteurs et des capacités MOS

I.3.1. La non-linéarité de la résistance de conduction

Considérons le montage suivant (Figure 1 - 67) qui comprend une capacité et un interrupteur NMOS (ou PMOS). Nous partons de l'hypothèse que la tension d'entrée V_e évolue en petits signaux autour du mode commun $V_e = V_{DD}/2 + \varepsilon$ et que la capacité est chargée à $V_c = V_{DD}/2$. Le NMOS est donc dans sa zone ohmique de fonctionnement et peut alors être remplacé par une résistance équivalente R_{ON} (résistance de conduction) :

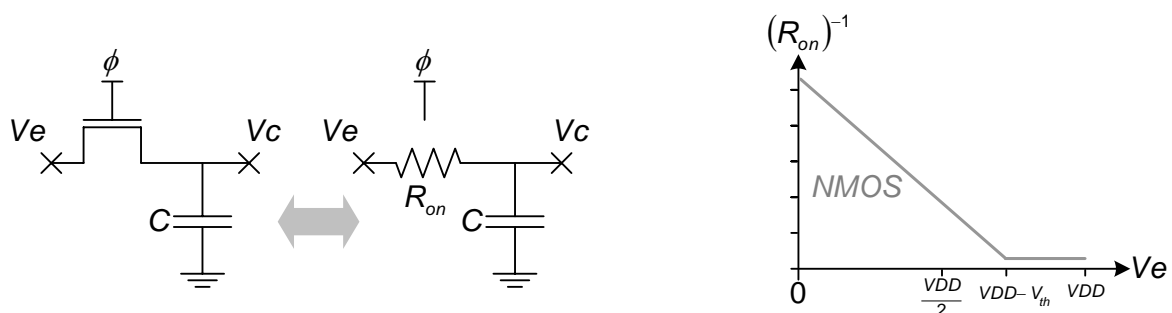


Figure 1 - 67 : Non-linéarité d'un interrupteur NMOS

La valeur de la résistance R_{ON} dépend alors de la tension à ses bornes [Bas00] [BBL96-2] :

$$(R_{ON})^{-1} = \mu_n \cdot C_{ox} \cdot \frac{W}{L} \cdot (V_{\phi_2} - V_e - V_{th}) \quad \text{Eq. 1 - 46}$$

Nous en déduisons donc que le temps de charge ou de décharge (proportionnels au produit $R_{ON} \cdot C$) dépend de la valeur de la tension d'entrée. Or pour avoir un échantillonnage efficace il faut que ce temps soit négligeable devant la période d'échantillonnage. Pour cela, nous fixons le produit $R_{ON} \cdot C$ au moins inférieur à un dixième de la période d'échantillonnage :

$$R_{ON} \cdot C \leq \frac{T_{ech}}{10} \quad \text{Eq. 1 - 47}$$

Or à fort signal (V_e proche de $V_{DD} - V_{th}$) la valeur de la résistance R_{ON} peut devenir infinie (l'interrupteur MOS est alors ouvert). Dans ce cas nous ne respectons plus l'équation précédente, ce qui peut induire des erreurs d'échantillonnage.

Il faut donc « linéariser » au maximum la fonction de transfert ou du moins éviter que l'interrupteur MOS sorte de sa zone ohmique de fonctionnement.

1.3.2. Technique de linéarisation de la résistance de conduction

La technique couramment utilisée pour « linéariser » les interrupteurs est l'utilisation de PMOS mis en parallèle avec le NMOS comme illustré par la Figure 1 - 68. Suivant les technologies employées, le PMOS est entre 3 et 5 fois supérieur en taille au NMOS. Les signaux de commande des MOS sont complémentaires (c.à.d. $V_{\phi} = VDD$ quand $V_{\bar{\phi}} = 0$ et inversement) [Bas00] :

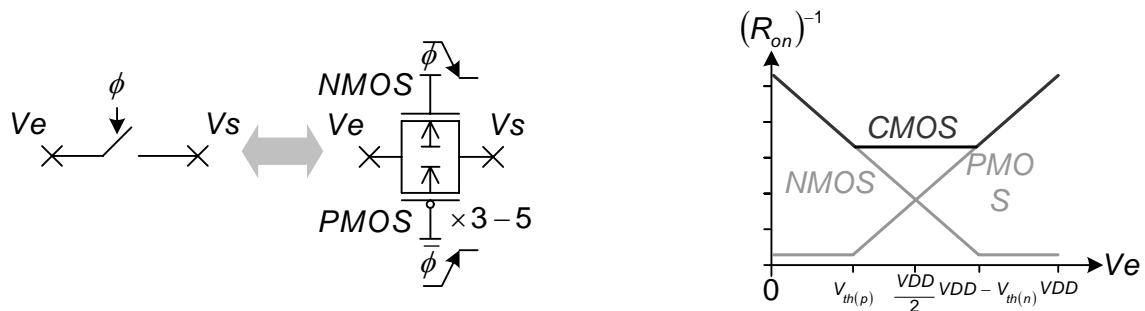


Figure 1 - 68 : Linéarisation par l'utilisation d'un interrupteur CMOS

1.3.3. Les phénomènes parasites d'injection d'horloge

1.3.3.1. *Les capacités parasites des interrupteurs MOS*

Une autre source d'erreur inhérente à ce type d'interrupteur provient des capacités parasites intrinsèques aux MOS. Plongeons nous dans la structure d'un MOS [Bas00] [BBL96-2].

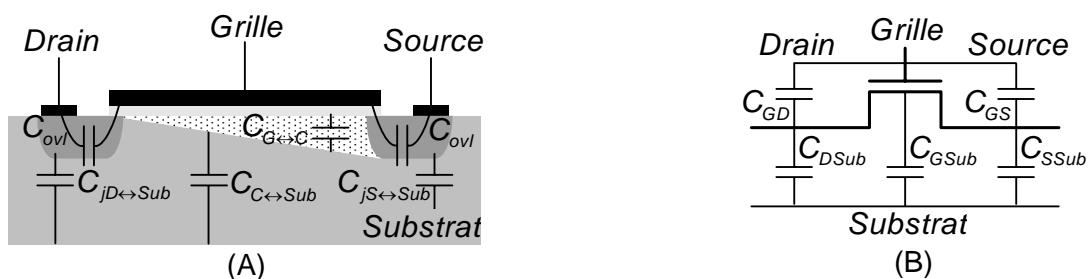


Figure 1 - 69 : Capacités parasites d'un transistor MOS

Comme illustré ci-dessus, le transistor MOS est composé de zones de dopage N ou P implantées dans un substrat de silicium dopé P. Il en découle deux capacités parasites $C_{jS \leftrightarrow Sub}$ et $C_{jD \leftrightarrow Sub}$ dites de jonction entre la Source, le Drain et le Substrat. L'oxyde de grille recouvre partiellement le drain et la source ce qui génère également des capacités de recouvrement (overlapping) notées C_{ovl} entre le drain, la source et la grille. La zone de déplétion où se forme le canal entre la source et le drain est

également le siège de capacités parasites $C_{G \leftrightarrow C}$ entre la grille et le canal et $C_{C \leftrightarrow Sub}$ entre le canal et le substrat. Ces capacités évoluent suivant la zone de fonctionnement du transistor et peuvent être regroupées comme indiqué dans la Figure 1 - 69 (B).

1.3.3.2. Le phénomène d'injection d'horloge

Reprenons le schéma composé d'un interrupteur NMOS et d'une capacité (cf. Figure 1 - 67). Dans notre zone de fonctionnement (ohmique) les capacités entre le drain, la source et la grille sont identiques ($C_{GD} = C_{GS} = C_{ovl}$) et la capacité entre la grille et le substrat nulle ($C_{GSub} = 0$). Etudions la coupe transversale suivante [Bas00] :

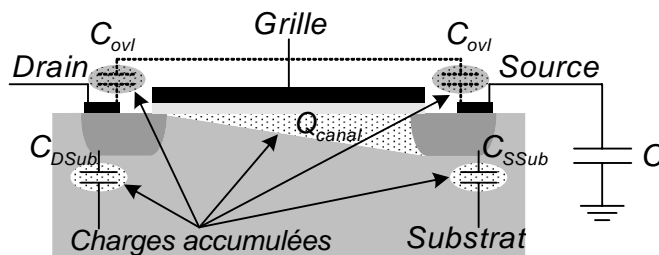


Figure 1 - 70 : Accumulation des charges dans un interrupteur MOS

Généralement seules les capacités de grille C_{GS} et C_{GD} ont une influence non négligeable sur le reste du circuit [Bas00]. Le MOS est passant, l'ensemble des capacités parasites est chargé.

Nous avons donc une charge « parasite » accumulée égale à :

$$Q_{parasite} = C_{ovl}(V_{GS} + V_{GD}) \quad \text{Eq. 1 - 48}$$

A cela s'ajoute les charges accumulées dans la zone de déplétion qui forme le canal :

$$Q_{canal} = C_{ox} \cdot W \cdot L \cdot (V_{GS} - V_{th}) \quad \text{Eq. 1 - 49}$$

Lorsque l'interrupteur s'ouvre ces charges sont redistribuées à travers le circuit, à travers le drain et la source comme illustré ci-dessous (Figure 1 - 71). Ce processus est connu sous le nom de phénomène d'injection d'horloge.

Cette redistribution entraîne une tension d'offset et donc une erreur sur la valeur de la tension aux bornes de la capacité de charge C égale à :

$$\Delta V_C = \frac{(k \cdot Q_{canal} + C_{ovl} \cdot V_{GS})}{(C + C_{ovl})} \quad \text{Eq. 1 - 50}$$

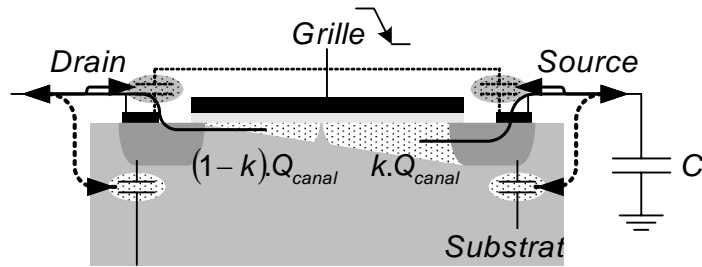


Figure 1 - 71 : Phénomène d'injection d'horloge

Dans notre cas nous considérons que l'ouverture de l'interrupteur s'effectue très longtemps après la stabilisation des tensions d'entrée et de sortie de sorte que le drain et la source sont au même potentiel ($VDD/2$) juste avant l'ouverture de l'interrupteur.

$$\Delta V_C = \frac{(k.C_{ox}.W.L(VDD/2 - V_{thN}) + C_{ovl}VDD)}{(C + C_{ovl})} \quad \text{Eq. 1 - 51}$$

Les capacités parasites étant une composante intrinsèque des transistors il est possible lors du layout de minimiser certaines capacités parasites comme la capacité de drain, mais pas de les annuler. Par contre il est possible d'annuler les injections de charges vues par la source et le drain, ou d'empêcher leur charge. Bien souvent on utilise les deux méthodes simultanément.

1.3.4. Techniques de compensation des phénomènes parasites

1.3.4.1. *Méthode utilisant les propriétés des PMOS*

La technique de compensation de charges la plus complète consiste à brancher un PMOS de même taille en parallèle avec le NMOS (les capacités parasites des PMOS et NMOS sont sensiblement de même valeur à taille de MOS égale). Les tensions de commande des MOS sont complémentaires, les injections de charges sont donc opposées et s'annulent [BBL96-2] :

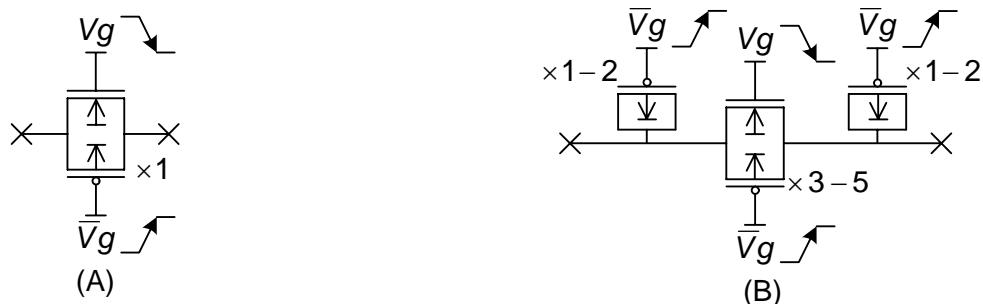


Figure 1 - 72 : Interrupteurs « compensés »

Cependant cette technique n'est pas compatible avec la technique de linéarisation des interrupteurs (cf. §1.3.2), à cause de la taille des PMOS mis en parallèle. Si l'on veut conserver et la linéarité et

une faible injection d'horloge on ajoute à l'interrupteur CMOS « linéarisé » deux autres PMOS montés en capacité (cf. figure ci-dessus (B)).

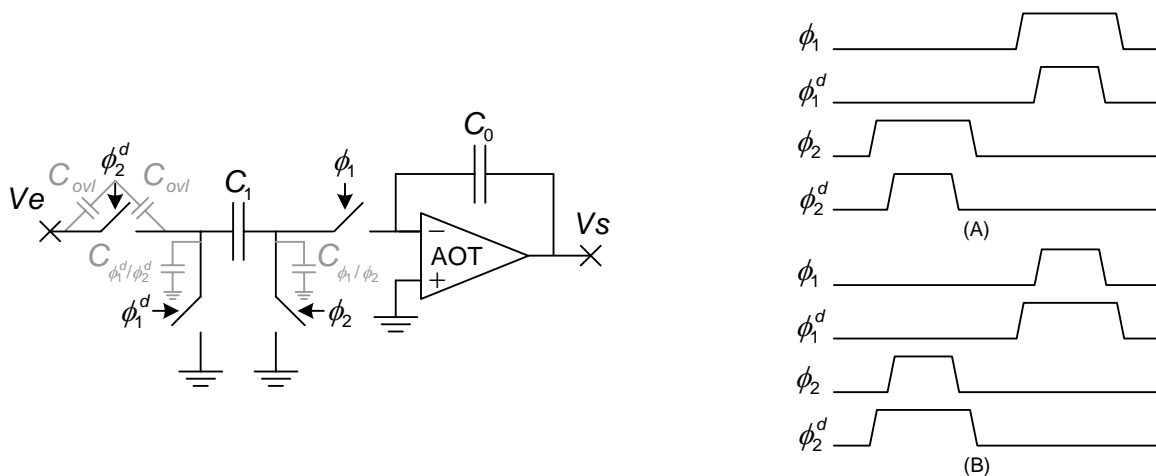
Cette solution, qui peut paraître idéale, ne doit être employée que dans les cas où le besoin de linéarité des interrupteurs est important, car elle peut s'avérer pire que le mal. En effet son utilisation entraîne la multiplication des MOS ce qui pose deux problèmes : premièrement le générateur de signaux de commande va devoir débiter un courant plus fort à chaque commutation pour charger les capacités de grille et devra donc être conçu en conséquence, deuxièmement la valeur des capacités C_{SSub} et C_{DSub} va augmenter, ce qui va nous poser un problème pour le transfert de charges d'une capacité à une autre. Nous allons expliquer ce dernier phénomène que dans le prochain paragraphe.

De plus il faut noter que cette compensation n'est valable que sur une certaine plage de tension (centrée autour de $VDD/2$ dans notre cas). En dehors de cette zone, la compensation se dégrade.

1.3.4.2. Méthode utilisant les chronogrammes

La seconde méthode est plus directement liée à l'utilisation des interrupteurs dans le montage à capacités commutées. Pour éviter la charge de certaines capacités parasites, il faut soit les court-circuiter en permanence, soit placer une de leurs bornes à un potentiel non défini. Pour cela les phases sont de nouveau dédoublées et un retard est introduit : $(\phi_1; \bar{\phi}_1)$, $(\phi_1^d; \bar{\phi}_1^d)$, $(\phi_2; \bar{\phi}_2)$ et $(\phi_2^d; \bar{\phi}_2^d)$.

Considérons le montage d'intégration positive (cf. Figure 1 - 73) sur lequel apparaissent les capacités parasites de recouvrement C_{ovl} du premier interrupteur ϕ_2^d ainsi que les capacités équivalentes Drain/Source Substrat C_{ϕ_1/ϕ_2} et $C_{\phi_1^d/\phi_2^d}$ des interrupteurs ϕ_1 , ϕ_2 , ϕ_1^d et ϕ_2^d [BBL96-2] :



Appliquons tout d'abord le chronogramme (A) au circuit :

Lorsque ϕ_2^d s'ouvre, les capacités parasites C_{ovl} de l'interrupteur ϕ_2^d vont se décharger respectivement vers l'entrée et vers le reste du circuit. La capacité parasite C_{ϕ_1/ϕ_2} étant court-circuitée et $C_{\phi_1^d/\phi_2^d} \ll C_1$, la majorité de la charge Q_{ovl} se retrouve alors dans C_1 .

Lorsque ϕ_2 s'ouvre, le nœud commun aux capacités C_1 et C_{ϕ_1/ϕ_2} est à un potentiel flottant il n'y a donc aucun transfert de charges, les équilibres électriques restent en l'état.

Cette variation de charge Q_{ovl} est ensuite intégralement transférée vers la capacité C_0 .

En appliquant le chronogramme (B) on ne rencontre plus ce type de problème :

En effet lorsque ϕ_2 s'ouvre, le nœud commun à C_1 et C_{ϕ_1/ϕ_2} est à un potentiel flottant il n'y a donc aucun transfert de charges, les équilibres électriques demeurent inchangés.

Puis ϕ_2^d s'ouvre. Les capacités parasites C_{ovl} se déchargent alors respectivement vers l'entrée et vers le reste du circuit. Leur charge se répartit donc dans les capacités C_{ϕ_1/ϕ_2} ,

$C_{\phi_1^d/\phi_2^d}$ et C_1 :

$$\left\{ \begin{array}{l} \Delta Q C_{\phi_1^d/\phi_2^d} \approx \frac{C_{\phi_1^d/\phi_2^d}}{C_{\phi_1^d/\phi_2^d} + C_{\phi_1/\phi_2}} Q_{Cov} \\ \Delta Q C_1 \approx \Delta Q C_{\phi_1/\phi_2} \approx \frac{C_{\phi_1/\phi_2}}{C_{\phi_1^d/\phi_2^d} + C_{\phi_1/\phi_2}} Q_{Cov} \end{array} \right. \quad \text{avec } (C_{\phi_1^d/\phi_2^d}; C_{\phi_1/\phi_2}) \ll C_1 \quad \text{Eq. 1 - 52}$$

Lorsque l'interrupteur ϕ_1^d se referme, l'ensemble des charges contenues dans la capacité $C_{\phi_1^d/\phi_2^d}$ partent à la masse.

Ainsi seule une quantité de charge $\Delta Q C_1 + \Delta Q C_{\phi_1/\phi_2}$ est transférée vers la capacité C_0 .

Cette variation de charge peut alors être minimisée si $C_{\phi_1^d/\phi_2^d} > C_{\phi_1/\phi_2}$. Nous allons montrer dans la sous-partie suivante que tel est le cas.

I.3.5. Les capacités parasites des capacités en poly-silicium

Nous a vu comment minimiser l'influence des éléments parasites des interrupteurs MOS, nous allons maintenant nous intéresser aux capacités MOS. En effet en étudiant la coupe d'une capacité MOS (Figure 1 - 74), nous constatons la présence, outre la capacité principale C_{AB} formée par les deux électrodes et le diélectrique en oxyde de silicium, de deux capacités parasites avec le substrat $C_{A \leftrightarrow Sub}$ et $C_{B \leftrightarrow Sub}$, soit le schéma équivalent suivant [BBL96-2] :

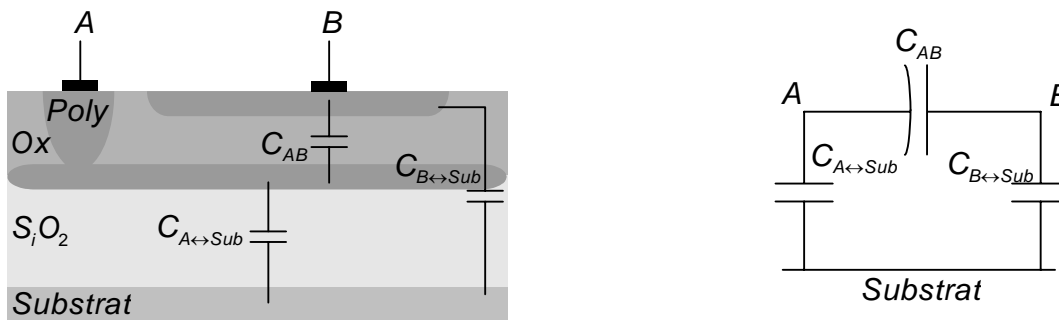


Figure 1 - 74 : Capacités parasites d'une capacité en poly-silicium

La solution pour minimiser la valeur de la capacité $C_{B \leftrightarrow Sub}$ technologique. En effet comme cela est représenté ci-dessus, l'électrode supérieure ne débord pas par rapport à l'électrode inférieure, bien au contraire, c'est l'inverse qui se produit. Cela réduit la valeur de cette capacité parasite et donc son influence que l'on peut considérer négligeable.

Cette méthode n'a aucune influence sur la capacité $C_{A \leftrightarrow Sub}$. En fait il n'existe pas de méthode dédiée à cette capacité, pour minimiser sa valeur et ses effets. Or il a été démontré précédemment (cf. §1.3.4.2) qu'il était intéressant d'avoir une capacité parasite plus forte en entrée de capacité qu'en sortie. La capacité est donc positionnée de façon à remplir cette condition.

II. ANNEXES : Chapitre II

II.1. Annexe 1 : Les sources de bruit composées

II.1.1. L'addition d'un bruit blanc et d'un bruit de Flicker

Considérons une source de bruit issue de l'addition de deux sources non-corrélées : une source de bruit blanc et une source de bruit de Flicker, définie dans le domaine fréquentiel par le spectre Figure 2 - 31. Nous noterons respectivement $S_{blanc}(f)_{dB_v/Hz}$, $S_{Flicker}(f)_{dB_v/Hz}$ et $S_b(f)_{dB_v/Hz}$ les DSP en dB_v/Hz du bruit blanc, du bruit de Flicker et du bruit sommé :

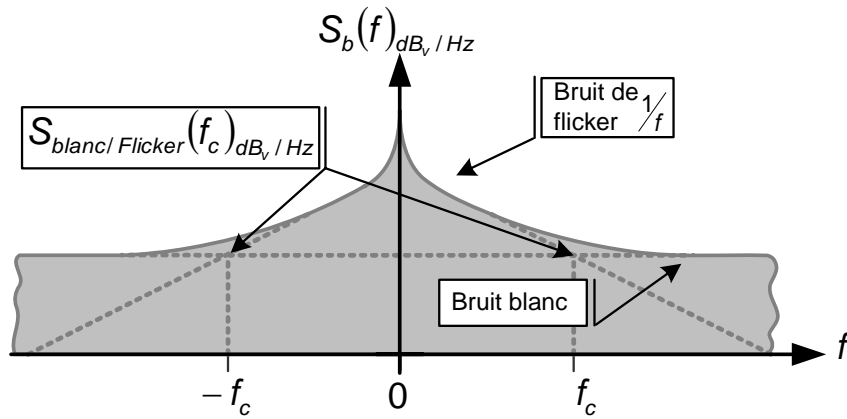


Figure 2 - 31 : Source de bruit blanc et de bruit de Flicker

Pour $-f_c < f < f_c$, le bruit a un comportement en $1/f$ (Flicker) et vaut $S_{Flicker}(f)_{dB_v/Hz}$ puis, pour $f < -f_c$ et $f > f_c$, le bruit est blanc, sa valeur est constante et vaut $S_{blanc}(\pm f_c)_{dB_v/Hz}$. Cette source sera donc définie par le couple : $(f_c; S_{blanc}(\pm f_c)_{dB_v/Hz})$. La fréquence d'intersection des deux tangentes est notée f_c (corner frequency, à ne pas confondre avec carrier frequency) et est définie par :

$$S_{Flicker}(\pm f_c)_{dB_v/Hz} = S_{blanc}(\pm f_c)_{dB_v/Hz} \quad \text{Eq. 2 - 85}$$

Comme les sources de bruits sont considérées indépendantes, nous pouvons sommer les puissances et écrire :

$$\begin{aligned} S_b(\pm f_c)_{V_{rms}^2/Hz} &= S_{blanc}(\pm f_c)_{V_{rms}^2/Hz} + S_{Flicker}(\pm f_c)_{V_{rms}^2/Hz} \\ &= 2 \cdot S_{blanc}(\pm f_c)_{V_{rms}^2/Hz} \quad (\text{ou } 2 \cdot S_{Flicker}(\pm f_c)_{V_{rms}^2/Hz}) \end{aligned} \quad \text{Eq. 2 - 86}$$

$$\text{Soit } \Rightarrow S_b(\pm f_c)_{dB_v/Hz} = 10 \cdot \log(2) + S_{blanc}(\pm f_c)_{dB_v/Hz} \quad (\text{ou } S_{Flicker}(\pm f_c)_{dB_v/Hz})$$

En résumé :

$$S_b(f)_{dB_v/Hz} = \begin{cases} S_{Flicker}(f)_{dB_v/Hz} & -f_c \ll f \ll f_c \\ S_{blanc}(\pm f_c)_{dB_v/Hz} + 10 \cdot \log(2) & f = \pm f_c \\ S_{blanc}(\pm f_c)_{dB_v/Hz} & f \gg f_c \text{ et } f \ll -f_c \end{cases} \quad \text{Eq. 2 - 87}$$

II.1.2. L'addition d'un bruit en $1/f^2$ et d'un bruit en $1/f^3$

Considérons une source de bruit issue de l'addition d'une source de bruit en $1/f^2$ et d'une source de bruit en $1/f^3$, définie dans le domaine fréquentiel par le spectre Figure 2 - 32. Nous noterons $S_{1/f^2}(f)_{dB_v/Hz}$, $S_{1/f^3}(f)_{dB_v/Hz}$ et $S_b(f)_{dB_v/Hz}$ les DSP en dB_v/Hz respectivement du bruit en $1/f^2$, du bruit en $1/f^3$ et du bruit somme :

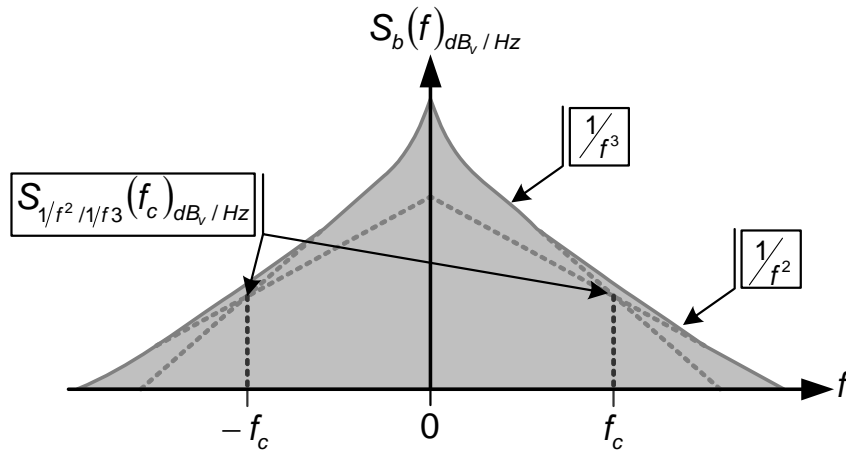


Figure 2 - 32 : Source de bruit en $1/f^2$ et de bruit en $1/f^3$

Pour $-f_c \ll f \ll f_c$ le bruit a un comportement en $1/f^3$ et vaut $S_{1/f^3}(f)_{dB_v/Hz}$ puis, pour $f < -f_c$ et $f > f_c$, le bruit a un comportement en $1/f^2$ et vaut $S_{1/f^2}(f)_{dB_v/Hz}$. Cette source sera donc définie par le couple : $(f_c; S_{1/f^2}(\pm f_c)_{dB_v/Hz})$. Comme précédemment la fréquence d'intersection des deux tangentes est notée f_c et est définie par :

$$S_{1/f^2}(\pm f_c)_{dB_v/Hz} = S_{1/f^3}(\pm f_c)_{dB_v/Hz} \quad \text{Eq. 2 - 88}$$

En résumé :

$$S_b(f)_{dB_v/Hz} = \begin{cases} S_{1/f^3}(f)_{dB_v/Hz} & -f_c \ll f \ll f_c \\ S_{1/f^2}(\pm f_c)_{dB_v/Hz} + 10 \cdot \log(2) & f = \pm f_c \\ S_{1/f^2}(f)_{dB_v/Hz} & f \gg f_c \text{ et } f \ll -f_c \end{cases} \quad \text{Eq. 2 - 89}$$

II.2. Annexe 2 : Modulation d'une source de bruit

Intéressons nous à la modulation par une porteuse f_o d'un signal $b(t)$ qui sera tour à tour égal à une sinusoïde, à une somme de sinusoïdes et à un bruit blanc de DSP en dB_V/Hz notée $S_b(f)_{dB_V/Hz}$. Dans tous les cas, la puissance totale P_b de $b(t)$ sera égale à : $P_b = \sigma^2$. Considérons le signal modulé suivant :

$$\begin{aligned} y_m(t) &= A.\sin(2.\pi.f_o.t) + A.\alpha.b(t).\sin(2.\pi.f_o.t) \\ &= \underbrace{x_o(t)}_{\text{porteuse idéale}} + \underbrace{\alpha.x_o(t).b(t)}_{\text{bruit additif}} \end{aligned} \quad \text{Eq. 2 - 90}$$

II.2.1. Modulation d'une sinusoïde

Nous considérons, dans un premier temps, que $b(t)$ est une sinusoïde.

$$b(t) = \sqrt{2}.\sigma.\sin(2.\pi.f_c.t) \quad \text{Eq. 2 - 91}$$

Sa DSP $S_b(f)$ est donc concentrée sur deux raies harmoniques :

$$S_b(f) = |B(f)|^2 = \sigma^2/2 . (\delta(f + f_c) + \delta(f - f_c)) \quad \text{Eq. 2 - 92}$$

$|B(f)|^2$ désigne la transformée de Fourier du signal $b(t)$.

Ici $S_b(f) = |B(f)|^2$ car $b(t)$ est déterministe et de puissance finie. En se rappelant que la Transformée de Fourier d'un produit de deux signaux déterministes est le produit de convolution des Transformées de Fourier de ces deux signaux, nous pouvons alors écrire que la DSP de $y_m(t)$ est égale à :

$$\begin{aligned} |Y_m(f)|^2 &= |X_o(f)|^2 + |M(f)|^2 \\ &= |X_o(f)|^2 + \alpha^2 . |B(f) * X_o(f)|^2 \\ &= \frac{A^2}{4} . (\delta(f + f_o) + \delta(f - f_o)) + \frac{(\alpha.A.\sigma)^2}{8} . (\delta(f + f_o + f_c) + \delta(f - f_o + f_c) \\ &\quad + \delta(f - f_o - f_c) + \delta(f - f_c + f_o)) \end{aligned} \quad \text{Eq. 2 - 93}$$

Où $|Y_m(f)|$, $|X_o(f)|$ et $|B(f)|$ désignent les transformées de Fourier de $y_m(t)$, $x_o(t)$ et $b(t)$.

La DSP de $y_m(t)$ se répartit alors :

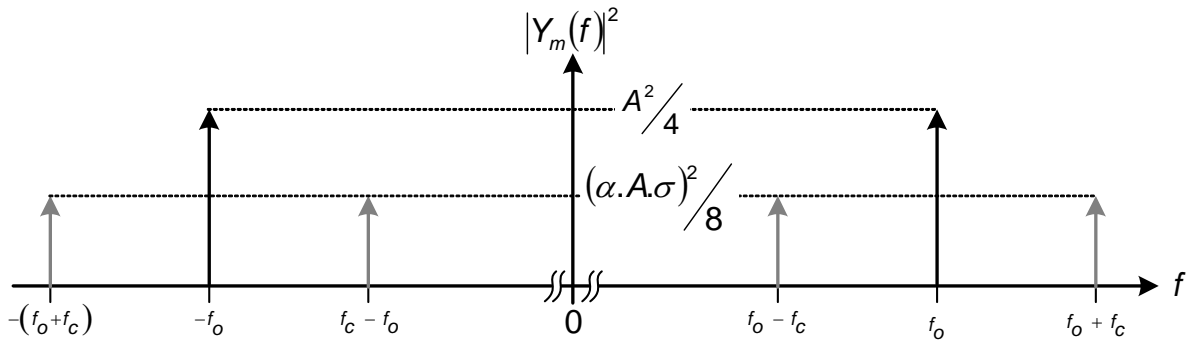


Figure 2 - 33 : DSP de $y_m(t)$

II.2.2. Modulation d'une somme finie de sinusoides

Nous considérons, dans un second temps, que $b(t)$ est la somme d'une valeur DC et de

$N \left(= \frac{f_b}{\Delta f = 1\text{Hz}} \right)$ sinusoides dont les fréquences sont espacées de $\Delta f = 1\text{Hz}$:

$$b(t) = \frac{\sigma}{\sqrt{2.N+1}} + \sum_{n=1}^{n=N} \frac{2.\sigma}{\sqrt{2.N+1}} . \sin(2.\pi.f_n.t) \quad \text{avec } f_n = n.\Delta f (= n.1\text{Hz}) \quad \text{Eq. 2 - 94}$$

La figure ci-dessous illustre le spectre de la DSP de $b(t)$, borné à l'intervalle de fréquence $[-f_b, f_b]$.

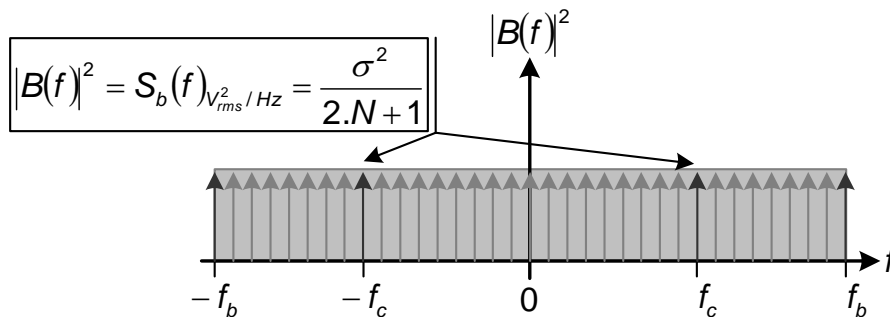


Figure 2 - 34 : DSP de $b(t)$

Nous supposons que $f_b \ll f_o$. Dans ce cas, les calculs réalisés précédemment (§II.2.1) restent valables. Nous considérons que la DSP du signal $b(t)$ est telle que :

$$|Y_m(\pm f_o)|^2 = X_o(\pm f_o) + \frac{\alpha^2.A^2}{4} . |B(0)|^2 = \frac{A^2}{4} + \frac{\alpha^2.A^2}{4} . \frac{\sigma^2}{2.N+1} \approx \frac{A^2}{4} \quad \text{Eq. 2 - 95}$$

La figure suivante illustre alors la DSP de $y_m(t)$:

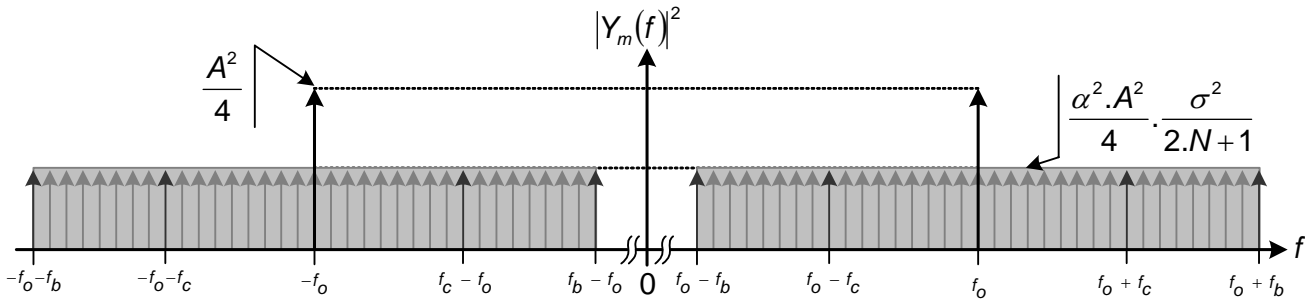


Figure 2 - 35 : DSP de $y_m(t)$

L'expression en dB_c/Hz la DSP de $y_m(t)$ rapportée à la puissance de la porteuse idéale $A^2/4$:

$$S_m(f)_{dBc/Hz} = \begin{cases} 10.\log\left(\alpha^2 \cdot \frac{\sigma^2}{2.N+1}\right) & \text{pour } f \in [-f_o - f_b, f_b - f_o] \text{ et } [f_o - f_b, f_o + f_b] \\ 0 & \text{sinon} \end{cases} \quad \text{Eq. 2 - 96}$$

N.B. : Ce raisonnement reste valable même si les sinusoïdes n'ont pas toutes la même amplitude : cf. la conclusion de cette annexe (§ II.2.4). Les choix effectués dans ce paragraphe s'inscrivent dans la continuité d'un raisonnement qui trouvera son dénouement dans le paragraphe suivant.

II.2.3. Modulation d'une somme infinie de sinusoïde

Nous considérons donc, dans un troisième temps, que $b(t)$ est une somme infinie de sinusoïdes. Sa DSP de bruit n'est plus bornée. Nous faisons donc tendre : ($f_b \rightarrow \infty, N \rightarrow \infty$). Nous assistons à un recouvrement des spectres des deux composantes du bruit modulé. Les raies de puissance se somment ce qui entraîne un doublement de la puissance :

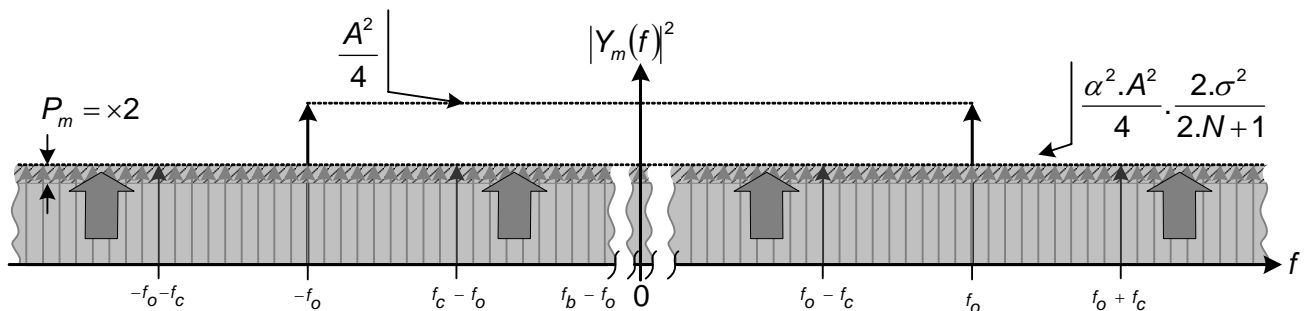


Figure 2 - 36 : DSP de $y_m(t)$

L'expression en dB_c/Hz de la DSP du bruit modulée est alors :

$$S_m(f)_{dB_c/Hz} = 10 \cdot \log \left(\alpha^2 \cdot \frac{2 \cdot \sigma^2}{2 \cdot N + 1} \right) \quad \text{pour } f \in [-\infty, +\infty] \quad \text{Eq. 2 - 97}$$

Cette expression peu paraître dénuée de sens étant donné que : $(f_b \rightarrow \infty, N \rightarrow \infty)$, néanmoins nous constatons une doublement de la puissance du bruit modulé qui entraîne une diminution de l'expression de la DSP en dB_c/Hz .

II.2.4. Conclusion

Si nous nous plaçons maintenant dans un contexte de simulation ou de mesure, c'est-à-dire de prélèvement d'échantillons et d'un traitement fréquentiel approprié, nous pouvons ramener l'ensemble de la puissance des sources de bruit dans un intervalle de fréquence : $[-F_{ech}/2, F_{ech}/2]$ et donc les quantifier ($\sigma^2/2 \cdot N + 1$ devient alors σ^2/F_{ech}). Nous pouvons alors distinguer deux catégories de bruits :

- ceux qui répondent aux conditions du paragraphe §II.2.2, c'est-à-dire ceux dont le spectre peut être considéré comme borné et qui donc ne se recouvrent pas dans une modulation, c'est le cas dans notre domaine d'étude des bruits de Flicker ($1/f$), en $1/f^2$ et en $1/f^3$.

L'expression en dB_c/Hz de la DSP du bruit modulée s'écrit alors :

$$S_m(\pm f_o \pm f)_{dB_c/Hz} = S_{Flicker}(\text{ou } 1/f^2 \text{ ou } 1/f^3)(\pm f)_{dB_v/Hz} + 10 \cdot \log(\alpha^2) \quad \text{Eq. 2 - 98}$$

- ceux qui répondent aux conditions du paragraphe §II.2.3, c'est-à-dire ceux dont le spectre n'est pas borné et qui donc se recouvrent lors de la modulation, c'est le cas du bruit blanc.

Dans ce cas l'expression en dB_c/Hz de la DSP du bruit modulée s'écrit :

$$S_m(\pm f_o \pm f)_{dB_c/Hz} = S_{blanc}(\pm f)_{dB_v/Hz} + 10 \cdot \log(2 \cdot \alpha^2) \quad \text{Eq. 2 - 99}$$

II.3. Annexe 3 : Modulation de sources composées

Cette annexe permet d'effectuer la synthèse des deux précédentes annexes.

II.3.1. Modulation d'une source de bruit issue de l'addition d'un bruit blanc et d'un bruit de Flicker

Nous considérons une source de bruit telle que celle décrite dans le paragraphe II.1.1 . Nous avons vu que la modulation d'une telle source entraîne la transposition du bruit autour du signal modulant et un chevauchement du bruit blanc entraînant le doublement de la puissance.

Nous effectuons une étude fréquentielle d'une telle source en ramenant dans l'intervalle de fréquences $[-F_{ech}/2, F_{ech}/2]$ l'ensemble de la puissance des bruits. Nous obtenons la figure suivante qui décrit l'ensemble des phénomènes :

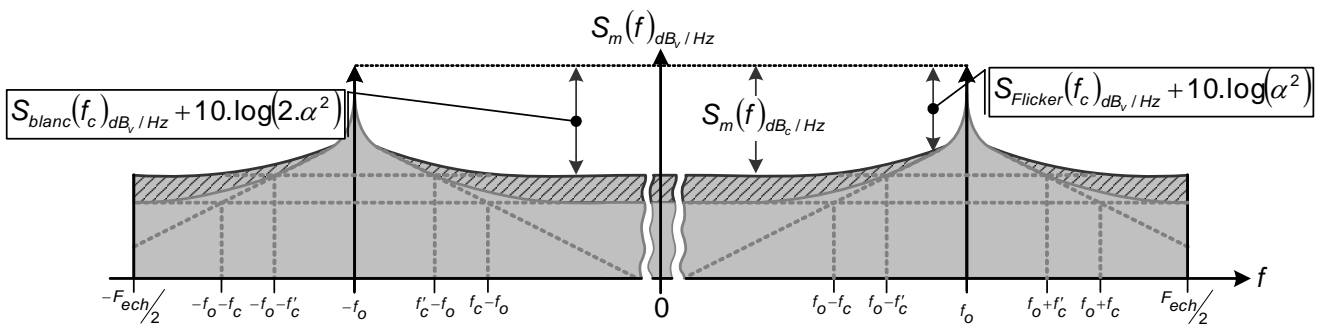


Figure 2 - 37 : DSP de $y_m(t)$

Le doublement de la puissance du bruit blanc modulé entraîne une nouvelle fréquence d'intersection avec le bruit de Flicker modulé. Nous la noterons f'_c .

Cette fréquence est définie comme suit :

$$\begin{aligned}
 S_{Flicker_modulé}(\pm f_o \pm f'_c)_{dBc/Hz} &= S_{blanc_modulé}(\pm f_o \pm f')_{dBc/Hz} \\
 S_{Flicker}(f')_{dBv/Hz} + 10.\log(\alpha^2) &= S_{blanc}(f')_{dBv/Hz} + 10.\log(2.\alpha^2) \\
 S_{Flicker}(f_c)_{dBv/Hz} + 10.\log(f_c/f'_c) &= S_{blanc}(f_c)_{dBv/Hz} + 10.\log(2)
 \end{aligned}
 \tag{Eq. 2 - 100}$$

Nous en déduisons la relation liant f_c et f'_c :

$$f'_c = f_c/2
 \tag{Eq. 2 - 101}$$

En tenant maintenant compte du décalage en fréquence, de la nouvelle fréquence de coupure, dus à la modulation du bruit autour de la porteuse, et de la somme des deux composantes en f'_c , nous pouvons écrire :

$$S_m(\pm f_o \pm f)_{dB_c/Hz} = \begin{cases} S_{Flicker}(\pm f)_{dB_v/Hz} + 10.\log(\alpha^2) & -f'_c < \pm f < f'_c \\ S_{Flicker // blanc}(\pm f_c)_{dB_v/Hz} + 10.\log(4.\alpha^2) & f = \pm f'_c \\ S_{blanc}(f_c)_{dB_v/Hz} + 10.\log(2.\alpha^2) & -f_o < \pm f < -f'_c / f'_c < \pm f < F_{ech}/2 \end{cases} \quad \text{Eq. 2 - 102}$$

II.3.2. Modulation d'une source de bruit issue de l'addition d'un bruit en $1/f^2$ et d'un bruit en $1/f^3$

Nous considérons une source de bruit telle que celle décrite dans le paragraphe II.1.2 . Nous avons vu que la modulation d'une telle source entraîne la transposition du bruit autour du signal modulant.

Nous effectuons une étude fréquentielle d'une telle source en ramenant dans l'intervalle de fréquences $[-F_{ech}/2, F_{ech}/2]$ l'ensemble de la puissance des bruits. Nous obtenons la figure suivante qui décrit ce phénomène :

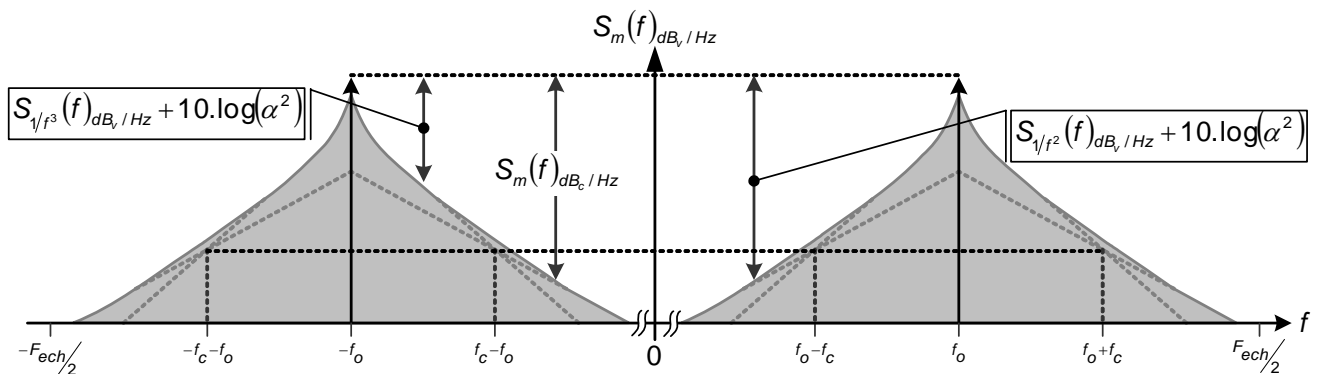


Figure 2 - 38 : DSP de $y_m(t)$

En tenant maintenant compte du décalage en fréquence, de la nouvelle fréquence de coupure, dus à la modulation du bruit autour de la porteuse, et de la somme des deux composantes en f'_c , nous pouvons écrire :

$$S_m(\pm f_o \pm f)_{dB_c/Hz} = \begin{cases} S_{1/f^3}(\pm f)_{dB_v/Hz} + 10.\log(\alpha^2) & -f'_c < \pm f < f'_c \\ S_{1/f^3 // 1/f^2}(\pm f_c)_{dB_v/Hz} + 10.\log(2.\alpha^2) & f = \pm f'_c \\ S_{1/f^2}(\pm f)_{dB_v/Hz} + 10.\log(\alpha^2) & -f_o < \pm f < -f'_c / f'_c < \pm f < F_{ech}/2 \end{cases} \quad \text{Eq. 2 - 103}$$

II.4. Annexe 4 : Synthèse d'un filtre d'ordre non-entier

II.4.1. Brique élémentaire

Considérons un filtre composé d'un filtre passe bas et d'un filtre passe haut ayant la fonction de transfert suivante :

$$\begin{aligned}
 H(f) &= H_{p.bas}(f) \cdot H_{p.haut}(f) \\
 &= \frac{H_{o.bas}}{1 + \tau \cdot j \cdot 2 \cdot \pi \cdot f} \cdot \frac{1 + \alpha \cdot \tau \cdot j \cdot 2 \cdot \pi \cdot f}{H_{o.haut}} \quad \text{avec} \quad \begin{cases} f_c = 1/2 \cdot \pi \cdot \tau \\ f'_c = 1/2 \cdot \pi \cdot \tau \cdot \alpha \end{cases} \quad \alpha < 1 \quad \text{Eq. 2 - 104} \\
 &= H_o \cdot \frac{1 + \alpha \cdot \tau \cdot j \cdot 2 \cdot \pi \cdot f}{1 + \tau \cdot j \cdot 2 \cdot \pi \cdot f}
 \end{aligned}$$

Si les deux fréquences de coupures (f_c et f'_c) sont suffisamment éloignées l'une de l'autre, la pente sur l'intervalle de fréquences $[f_c; f'_c]$ sera de -20 dB/décade . Cependant, en jouant sur la position de f_c et f'_c , et donc sur la valeur de α , on peut obtenir des pentes inférieures à -20 dB/décade , comme cela est illustré par la figure ci-dessous :

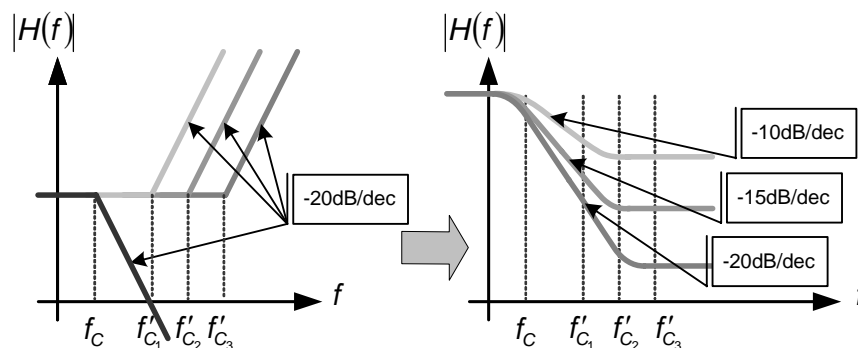


Figure 2 - 39 : Filtre $H(f)$ et différentes pentes

Nous allons chercher la (ou les) valeur(s) de α qui nous permettra d'obtenir un filtre avec une pente de -10 dB/décade (ou -3 dB/octave). Afin de quantifier la pente, nous introduisons un point de test, autour duquel nous calculerons la pente. Nous définissons ainsi la fréquence milieu f_m par :

$$f_m = \sqrt{f_c \cdot f'_c} = \frac{1}{2 \cdot \pi \cdot \sqrt{\alpha \cdot \tau}} \quad \text{Eq. 2 - 105}$$

Nous considérons maintenant que la fonction $H(f)$ est une droite affine sur l'intervalle de fréquence $[f_m/\beta; \beta \cdot f_m]$ centré autour de la fréquence milieu f_m .

La pente en ce point est alors égale à :

$$\begin{aligned} \Delta_{\left[\frac{f_m}{\beta}; \beta \cdot f_m\right]} &= 10 \cdot \log\left(\left|H(\beta \cdot f_m)\right|^2\right) - 10 \cdot \log\left(\left|H\left(\frac{f_m}{\beta}\right)\right|^2\right) \\ &= 10 \cdot \log\left(H_o \cdot \frac{1 + \alpha\beta^2}{1 + \beta^2/\alpha}\right) - 10 \cdot \log\left(H_o \cdot \frac{1 + \alpha/\beta^2}{1 + 1/\alpha\beta^2}\right) \\ &= 10 \cdot \log\left(\left(\frac{1 + \alpha\beta^2}{\alpha + \beta^2}\right)^2\right) \end{aligned} \quad \text{Eq. 2 - 106}$$

Si on veut que le module ait un comportement en $1/f$, il faut que la pente en ce point vérifie :

$$\begin{aligned} \Delta_{\left[\frac{f_m}{\beta}; \beta \cdot f_m\right]} &= 10 \cdot \log\left(\frac{1}{\beta \cdot f_m}\right) - 10 \cdot \log\left(\frac{1}{f_m/\beta}\right) \\ &= 10 \cdot \log\left(\frac{1}{\beta^2}\right) \end{aligned} \quad \text{Eq. 2 - 107}$$

Nous en déduisons alors la relation :

$$\alpha = \frac{\beta}{\beta^2 + \beta + 1} \quad \text{Eq. 2 - 108}$$

Considérons l'exemple suivant où $\beta = 2$ ($\alpha \approx 0.286$), $f_m = 100\text{Hz}$ et $H_o = 1$:

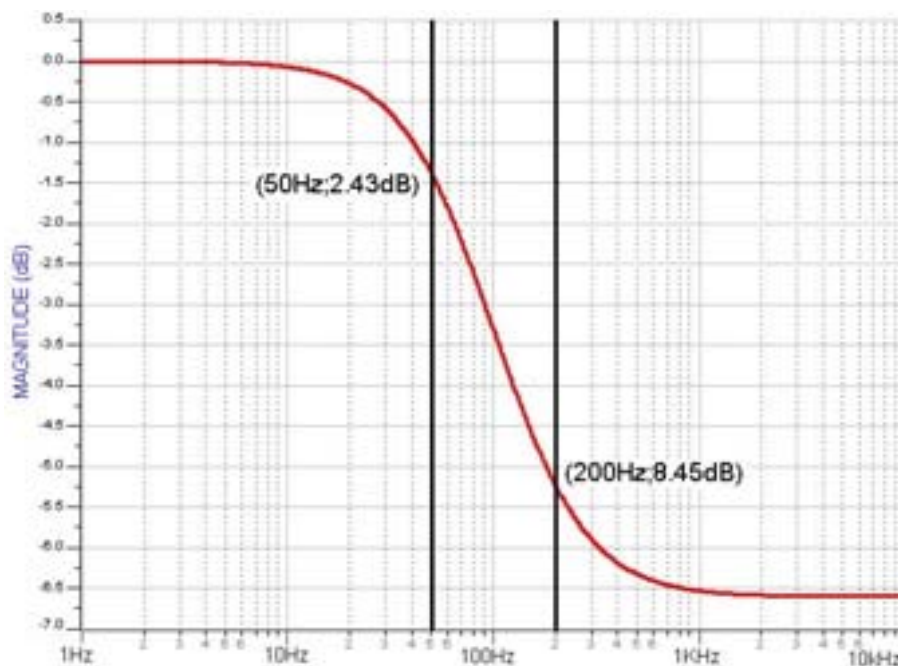


Figure 2 - 40 : Filtre $H(f)$

Le tableau ci-dessous nous donne la pente de $H(f)$ en fonction des intervalles de fréquences centrés autour de la fréquence f_m :

Pente (dB)	$\left[\frac{f_m}{\sqrt{2}}; \sqrt{2}.f_m \right]$	$\left[\frac{f_m}{2}; 2.f_m \right]$	$\left[\frac{f_m}{\sqrt{8}}; \sqrt{8}.f_m \right]$	$\left[\frac{f_m}{\sqrt{10}}; \sqrt{10}.f_m \right]$
$\beta = 2$	-3.25	-6.02	-8.03	-8.52
Théorie	-3.01	-6.02	-9.03	10.0

Tab. 2 - 1

Nous constatons alors que la pente n'est valable que sur l'intervalle de fréquences $\left[\frac{f_m}{2}; 2.f_m \right]$ et que, au-delà $\left[\frac{f_m}{\sqrt{2}}; \sqrt{2}.f_m \right]$; $\left[\frac{f_m}{\sqrt{8}}; \sqrt{8}.f_m \right]$ ou $\left[\frac{f_m}{\sqrt{10}}; \sqrt{10}.f_m \right]$, elle n'est plus conforme à la théorie.

II.4.2. Filtre complet

Afin d'avoir une bande de fréquence plus large, nous cascadons plusieurs filtres analogues au précédent en les décalant du facteur γ sur l'axe des fréquences. On définit alors un nouveau filtre par :

$$H(f) = \prod_{j=0}^{j=n} H_j(f) \text{ où } H_i(f) = H_o^i \frac{1 + j.2.\pi.f.\alpha.\tau_i}{1 + j.2.\pi.f.\tau_i} \text{ avec } \tau_i = (2.\pi.\sqrt{\alpha}.f_m^i)^{-1} \quad \text{Eq. 2 - 109}$$

La figure suivante illustre cette mise en cascade des filtres :

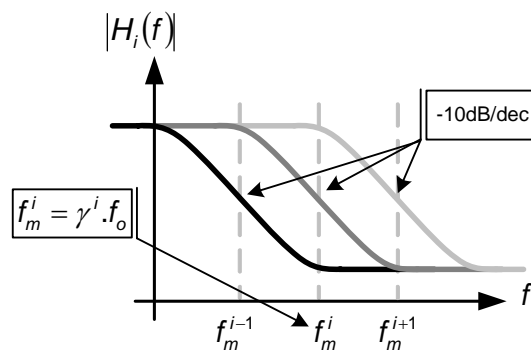


Figure 2 - 41 : Fonctions $H_i(f)$ avec une pente de -10 dB/décade sur l'intervalle $\left[\frac{f_m^i}{\beta}; \beta.f_m^i \right]$

Le calcul de la pente entre deux fréquences milieu f_m^i et f_m^{i+1} est le suivant :

$$\begin{aligned}
 \Delta_{[f_m^i, f_m^{i+1}]} &= 10.\log\left(\prod_{j=0}^{j=n} H_j(f_m^{i+1})\right) - 10.\log\left(\prod_{j=0}^{j=n} H_j(f_m^i)\right) \\
 &= 10.\log\left(\prod_{j=0}^{j=n} H_o^i \cdot \frac{1 + \alpha \cdot \gamma^{2(i+1-j)}}{1 + \gamma^{2(i+1-j)} / \alpha}\right) - 10.\log\left(\prod_{j=0}^{j=n} H_o^i \cdot \frac{1 + \alpha \cdot \gamma^{2(i-j)}}{1 + \gamma^{2(i-j)} / \alpha}\right) \\
 &= 10.\log\left(\frac{1 + \alpha \cdot \gamma^{2i}}{1 + \gamma^{2i} / \alpha}\right) - 10.\log\left(\frac{1 + \alpha / \gamma^{2(n-1)}}{1 + 1 / \alpha \cdot \gamma^{2(n-1)}}\right) \\
 &\approx 10.\log(\alpha^2)
 \end{aligned}$$

Eq. 2 - 110

On obtient alors une première relation liant γ et α :

$$\begin{aligned}
 \Delta_{[f_m^i, f_m^{i+1}]} &= 10.\log\left(\frac{1}{\gamma}\right) \approx 10.\log(\alpha^2) \\
 \Rightarrow \alpha^2 &= \frac{1}{\gamma}
 \end{aligned}$$

Eq. 2 - 111

Comme illustré par la Figure 2 - 41, la pente en f_m^i ne dépend plus uniquement de $H_i(f)$ mais également des filtres $H_{i-1}(f)$ et $H_{i+1}(f)$. Il faut donc recalculer la pente et, comme précédemment (Eq. 2 - 107), il faut vérifier :

$$\begin{aligned}
 \Delta_{\left[\frac{f_m^i}{\beta}, \beta \cdot f_m^i\right]} &= 10.\log\left(|H_{i-1}|^2 |H_i|^2 |H_{i+1}|^2 (\beta \cdot f_m^i)\right) - 10.\log\left(|H_{i-1}|^2 |H_i|^2 |H_{i+1}|^2 \left(\frac{f_m^i}{\beta}\right)\right) \\
 &= 10.\log\left(\left(\frac{1 + \alpha\beta^2}{\alpha + \beta^2}\right)^2 \left(\frac{1 + \alpha\gamma^2\beta^2}{\alpha + \gamma^2\beta^2}\right)^2 \left(\frac{\gamma^2 + \alpha\beta^2}{\alpha\gamma^2 + \beta^2}\right)^2\right) \\
 &\approx 10.\log\left(\frac{1}{\beta^2}\right)
 \end{aligned}$$

Eq. 2 - 112

On obtient alors un polynôme en α du 10ième ordre :

$$\begin{aligned}
 \alpha\beta^3(1-\beta) + \alpha^2\beta^2(\beta^3-1) + \alpha^4\beta(1-\beta^5) + \alpha^5\beta^3(1-\beta) + \alpha^6\beta^2(\beta^3-1) + \alpha^7(\beta^7-1) \\
 + \alpha^9\beta^3(1-\beta) + \alpha^{10}\beta^2(\beta^3-1) = 0
 \end{aligned}$$

Eq. 2 - 113

Nous avons effectué la résolution de manière graphique pour différentes valeurs de β^2 :

β^2	2	3	4	5	6	7	8
α	467.77m	445m	422.18m	400.29m	379.51m	360.06m	342.14m
γ	4.57	5.05	5.61	6.24	6.94	7.71	8.54

Tab. 2 - 2

Donc, pour couvrir l'intervalle de fréquences $[100\text{Hz}, 1\text{MHz}]$, il faut prendre :

β^2	2	3	4	5	6	7	8
Nombre de filtres	7	7	7	6	6	6	5

Tab. 2 - 3

La figure suivante illustre un filtre issu de la mise en cascade de 7 filtres avec les paramètres : $\beta^2 = 2$, $\alpha = 467.77m$ et $\gamma = 4.57$:

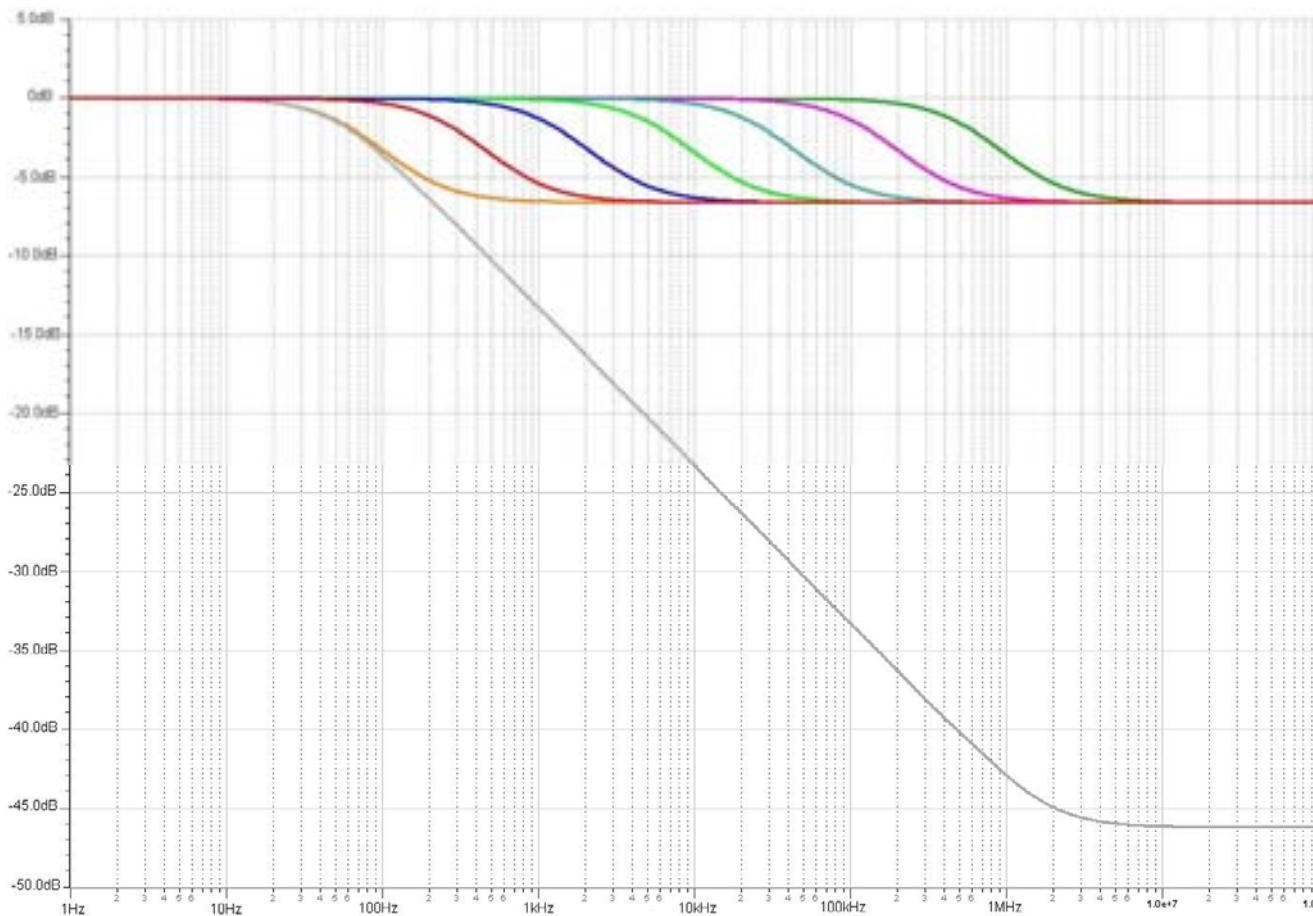


Figure 2 - 42 : Fonction $H(f)$ avec une pente moyenne de -10 dB/décade sur $[100\text{Hz}, 1\text{MHz}]$

II.5. Annexe 5 : Courbes sur l'intervalle $[-F_{ech}/2, F_{ech}/2]$

II.5.1. Bruit de Flicker et bruit blanc

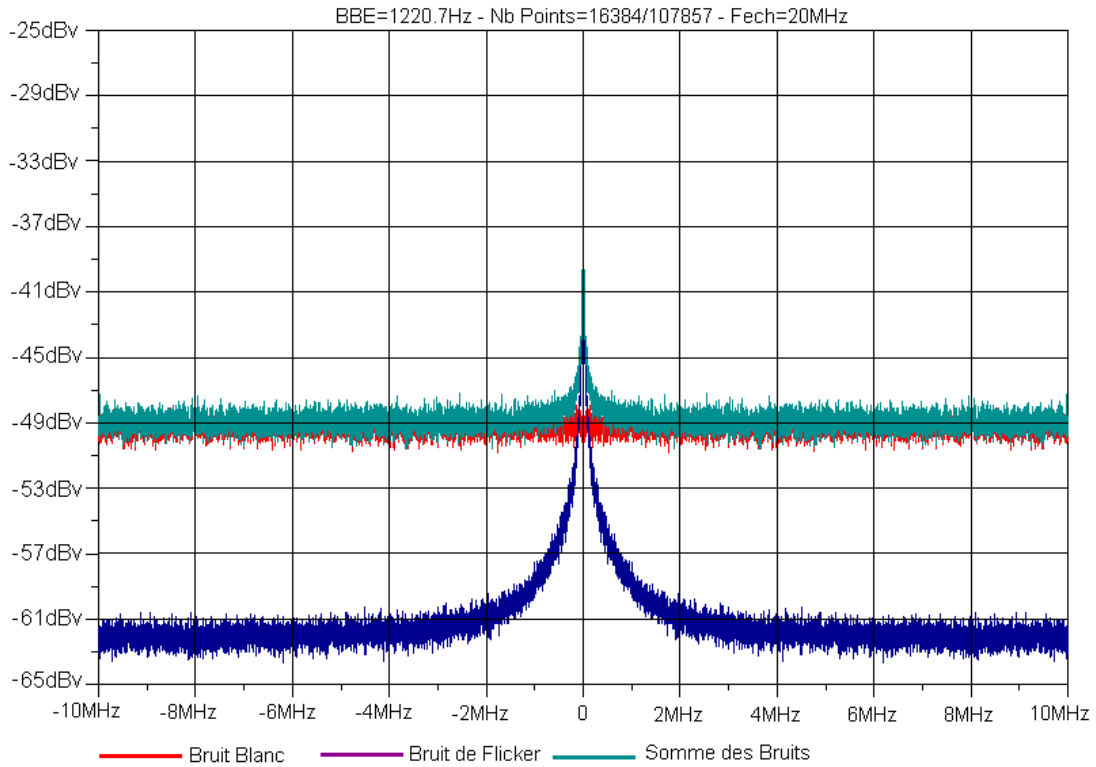


Figure 2 - 43 : Spectre du bruit de Flicker et du bruit blanc

II.5.2. Bruit de phase

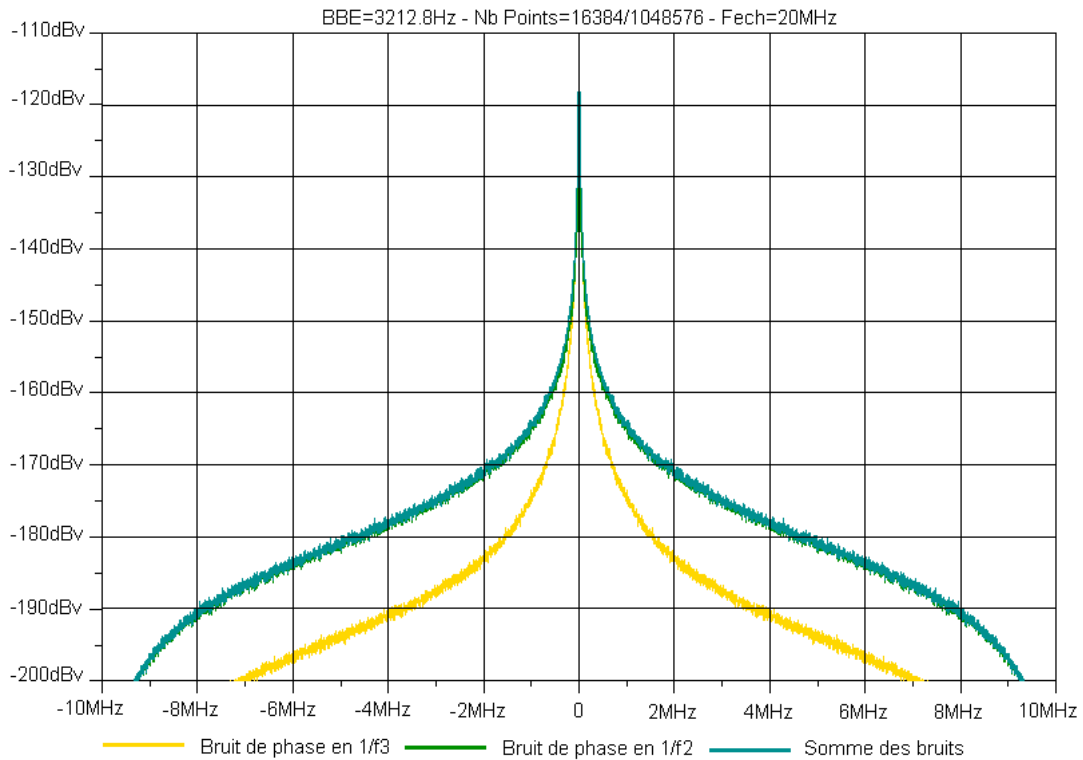


Figure 2 - 44 : Spectre du bruit de phase du générateur de sinusoides

II.5.3. Sinusoïde bruitée

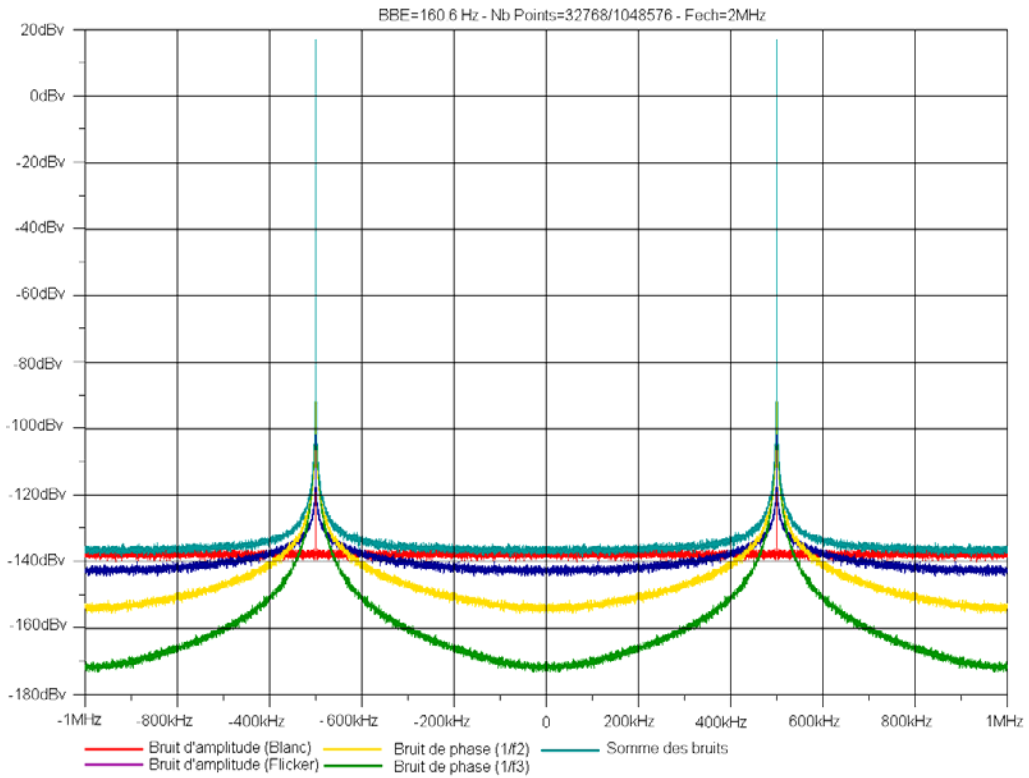


Figure 2 - 45 : Spectre du bruit de phase du générateur de sinusoïdes

II.5.4. Echantillonnage bruité

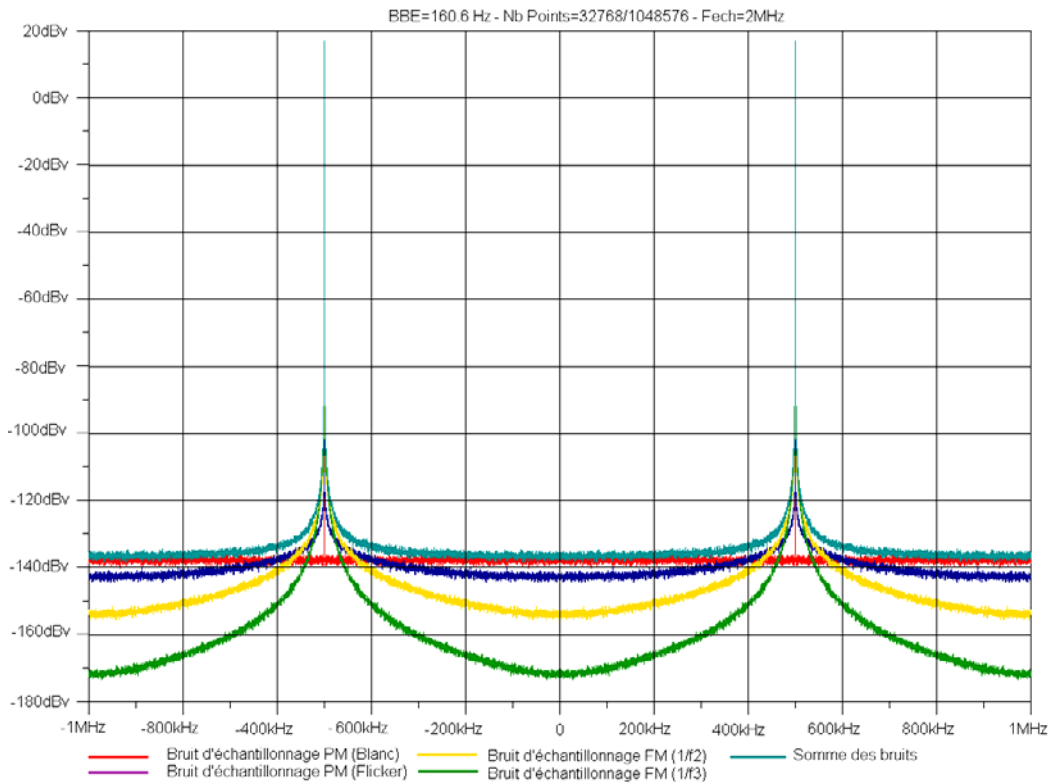


Figure 2 - 46 : Spectre du bruit de phase d'un échantillonnage jitté (FM et PM jitter)

III. ANNEXES : Chapitre III

III.1. Annexe 1 : Layout du circuit « Gain_10 »

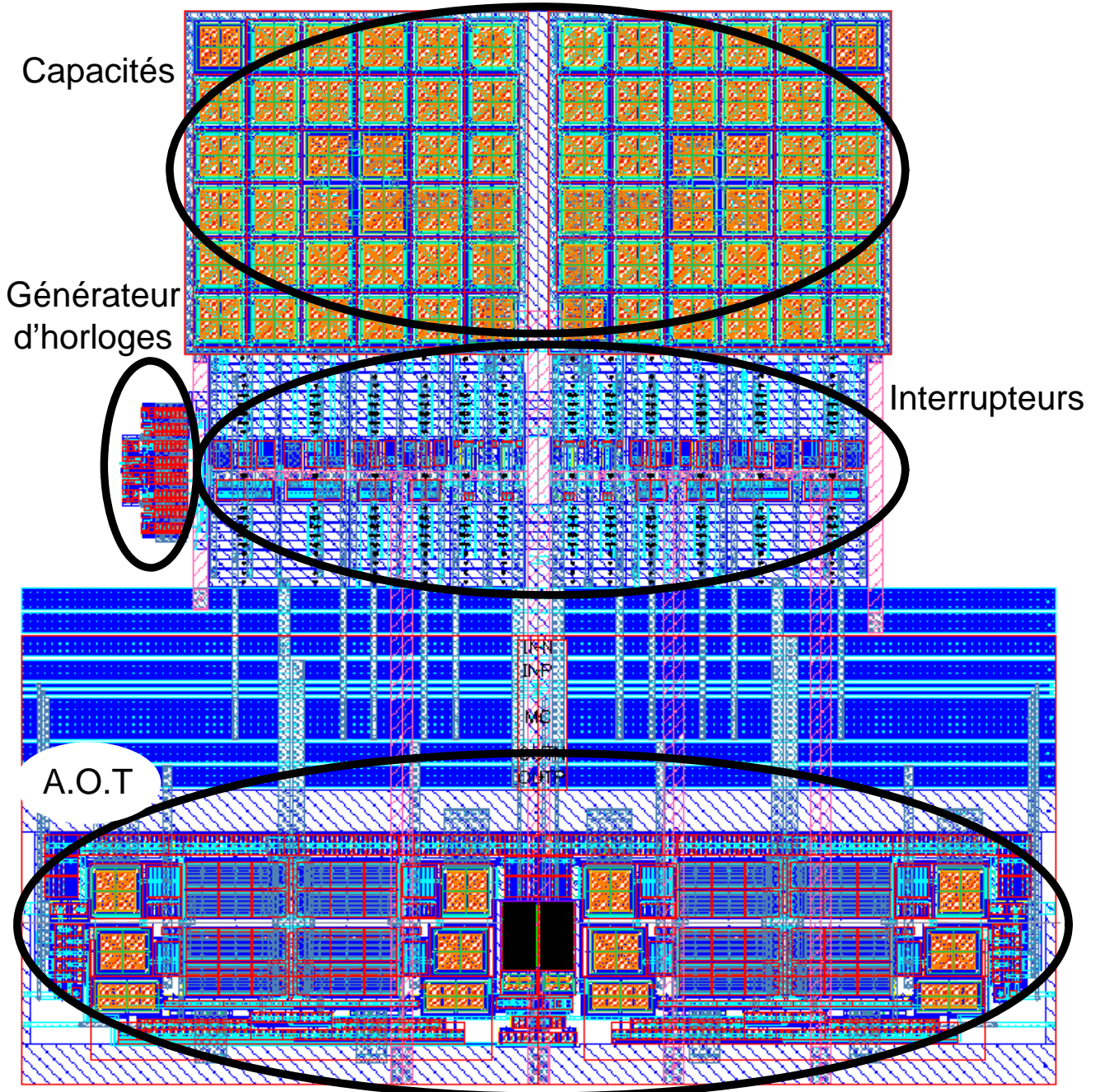


Figure 3 - 18 : Layout du circuit « Gain_10 »

III.2. Annexe 2 : Approximation de l'écart type de $\alpha.A.2.\pi.F_{ech} \cdot \cos(\phi_e) \sum_{i=0}^{i=n-1} \beta^i \cdot \delta_{total}^i$

Nous tenterons dans cette partie de trouver une relation entre les écarts types de la somme $\alpha.A.2.\pi.F_{ech} \cdot \cos(\phi_e) \sum_{i=0}^{i=n-1} \beta^i \cdot \delta_{total}^i$ et de la variable aléatoire δ_{total} .

III.2.1. Insuffisance de l'approche théorique

Nous réalisons dans cette partie une étude théorique en partant des caractéristiques statistiques du jitter total. Tout au long de cette étude nous illustrerons nos formules par des histogrammes réalisés sous SCILAB [SCI].

Considérons notre variable aléatoire qu'est le jitter δ_{total} , sa loi de répartition statistique $P_{\delta_{total}}(u)$ est une gaussienne de valeur moyenne m_{total} et d'écart type σ_{total} :

$$P_{\delta_{total}}(u) = \frac{1}{\sigma_{total} \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(u - m_{total})^2}{2 \sigma_{total}^2}} \quad \text{Eq. 3 - 20}$$

Nous présentons dans l'exemple ci-dessous l'historgramme issu du tirage de $10e^3$ échantillons avec $m_{total} = 0$ et $\sigma_{total} = 3.16e^{-4}$:

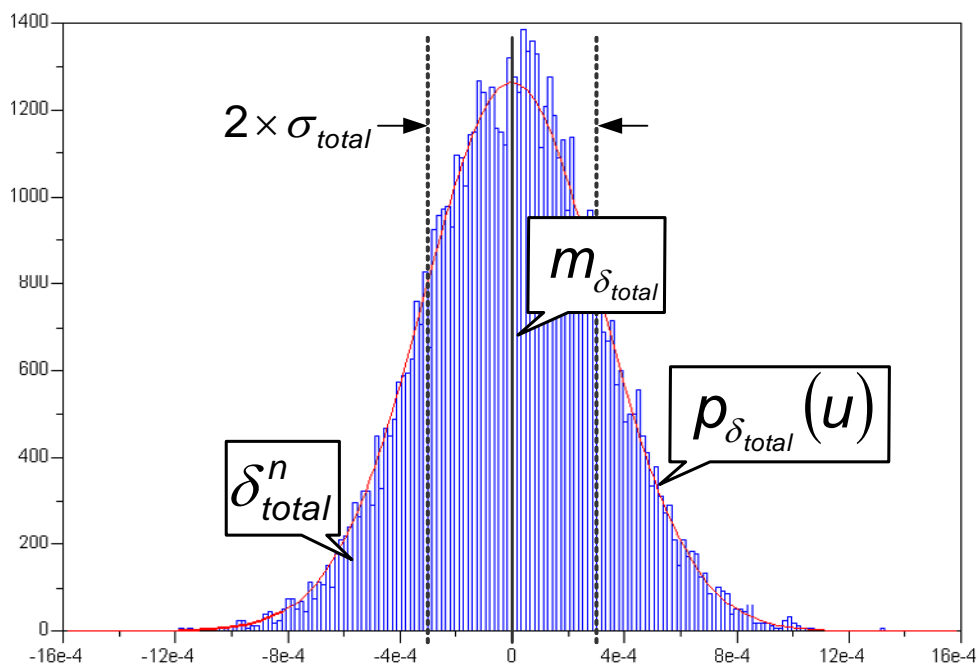


Figure 3 - 19 : Répartition statistique de δ_{total}

Considérons maintenant la variable aléatoire e'_n définie par la relation suivante :

$$\begin{aligned} e'_n &= A \cdot \sin(2 \cdot \pi \cdot F_{ech} \cdot (n \cdot T_{ech} + \delta_{total}^n) + \phi_e) \\ &= A \cdot \sin(2 \cdot \pi \cdot F_{ech} \cdot \delta_{total}^n + \phi_e) \end{aligned} \quad \text{Eq. 3 - 21}$$

Nous savons que :

$$\text{si } Y = a \cdot X + b \text{ alors } P_Y(u) = \frac{1}{|a|} \cdot P_X\left(\frac{u-b}{a}\right) \quad \text{Eq. 3 - 22}$$

De plus :

$$\text{si } Z = c \cdot \sin(d \cdot Y) \text{ alors } P_Z(u) = \frac{1}{d} \cdot \frac{1}{\sqrt{c^2 - u^2}} \cdot P_Y\left(\arcsin\left(\frac{u}{c}\right)\right) \quad \text{Eq. 3 - 23}$$

Nous en déduisons donc que la fonction de répartition de e'_n s'écrit alors :

$$P_{e'_n}(u) = \frac{1}{\sigma_{total} \sqrt{2 \cdot \pi}} \cdot \frac{1}{2 \cdot \pi \cdot F_{ech}} \cdot \frac{1}{\sqrt{A^2 - u^2}} e^{-\frac{\left(\frac{1}{2 \cdot \pi} \arcsin\left(\frac{u}{A}\right) - m_{total} - \frac{\phi_e}{2 \cdot \pi}\right)^2}{2 \sigma^2}} \quad \text{Eq. 3 - 24}$$

Nous présentons dans l'exemple ci-dessous l'histogramme issu du tirage de $10e^3$ échantillons avec $A = 0.02$, $\phi_e = 0$, $m_{total} = 0$ et $\sigma_{total} = 3.16e^{-4}$:

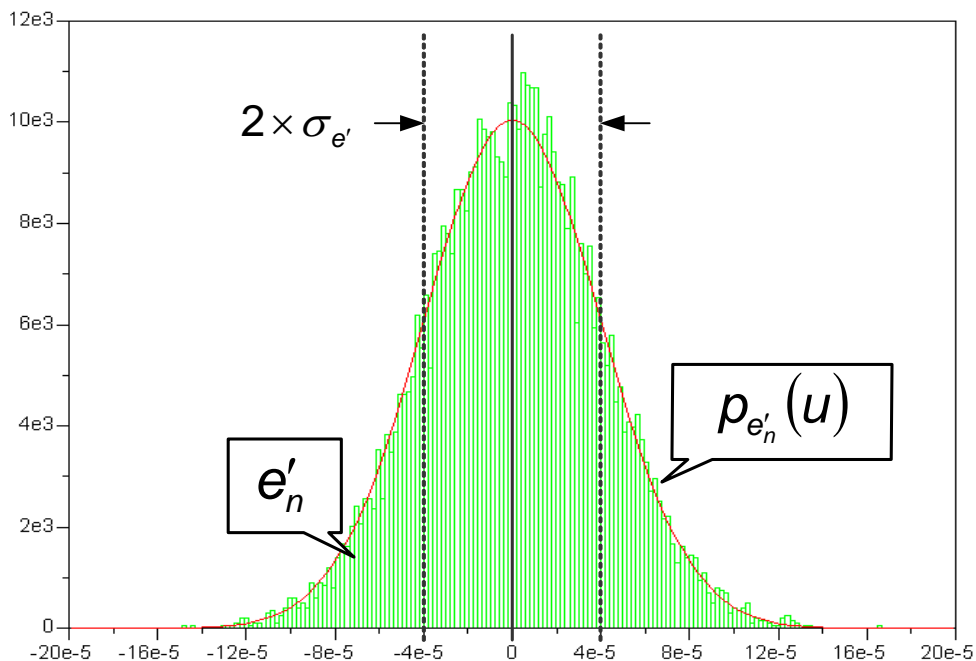


Figure 3 - 20 : Répartition statistique de e'_n

N.B. : En nous plaçant dans le cas qui nous intéresse où $\phi_e = 0, m_{total} = 0$ et en faisant l'approximation suivante :

$$e'_n \approx 2.\pi.A.F_{ech}.\delta_{total}^n \text{ car alors } e_n = 0 \quad \text{Eq. 3 - 25}$$

Alors la nouvelle fonction de répartition statistique devient :

$$P_{e'_n}^*(u) = \frac{1}{2.\pi.A.\sigma_{total}.F_{ech}.\sqrt{2.\pi}} . e^{\frac{(u)^2}{2(2.\pi.A.F_{ech}.\sigma_{total})^2}} \quad \text{Eq. 3 - 26}$$

Si nous traçons la différence entre les deux fonctions de répartition statistique pour les valeurs qui nous intéressent $A = 0.02$ et $\sigma_{total} = 3.16e^{-4}$ nous trouvons :

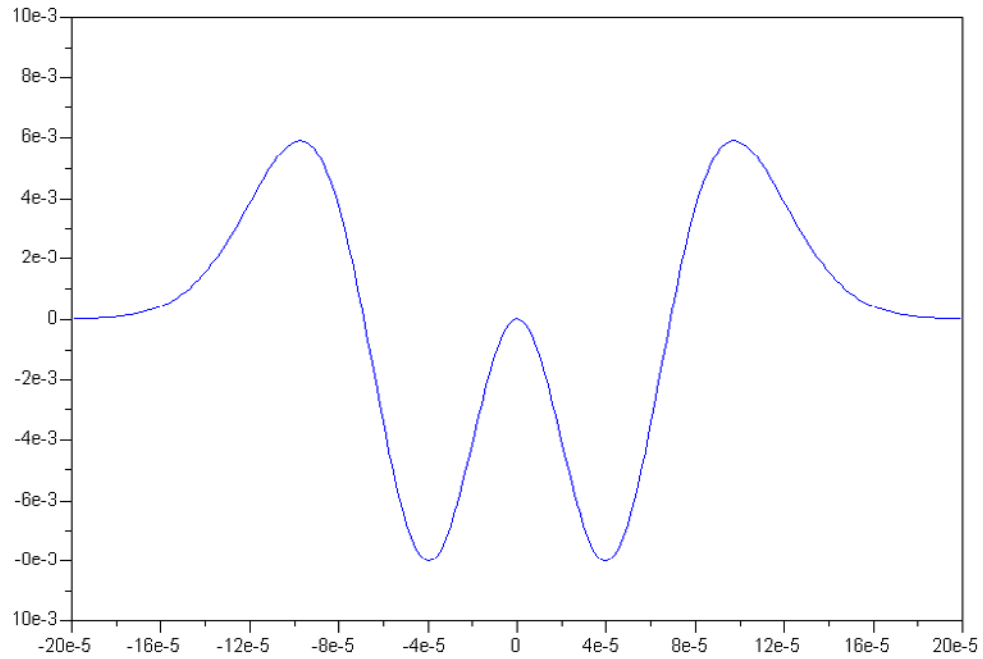


Figure 3 - 21 : Différence entre $P_{e'_n}^*(u) - P_{e'_n}(u)$

Au regard de la faible erreur engendrée par cette approximation nous allons l'utiliser par la suite dans notre étude pour déterminer les caractéristiques statistiques de la sortie du filtre.

Considérons maintenant la variable aléatoire s'_n définie par la relation suivante :

$$s'_n = \alpha . \sum_{i=0}^{i=n-1} \beta^i . A . \sin(2.\pi.F_{ech}.\delta_{total}^{n-i-1} + \phi_e) \quad \text{Eq. 3 - 27}$$

Soit pour $\phi_e = 0, m_{total} = 0$:

$$s'_n \approx \alpha . A . 2.\pi . F_{ech} . \sum_{i=0}^{i=n-1} \beta^i . \delta_{total}^{n-i-1} \quad \text{Eq. 3 - 28}$$

Ici s'arrête notre étude théorique, car malgré nos recherches, nous sommes dans l'incapacité de déterminer la loi de répartition statistique de cette fonction. Nous allons donc devoir effectuer, de

nouveau, une approximation en considérant que cette loi est de type gaussienne. Pour déterminer l'écart type nous allons alors effectuer une simulation du filtre puis réaliser une étude statistique de la sortie du filtre. Nous utiliserons alors les formules d'estimation suivantes :

$$m_{s'} = \frac{1}{N} \cdot \sum_{i=1}^{i=N} s'_i$$

$$\sigma_{s'} = \frac{1}{N} \cdot \sum_{i=1}^{i=N} (s'_i - m_{s'})^2$$

Eq. 3 - 29

III.2.2. Simulation d'un modèle idéal de filtre

Nous avons effectué la première simulation sous SCILAB [SCI] en utilisant un modèle idéal du filtre c'est-à-dire en simulant l'Eq. 3 - 27. Nous trouvons alors l'histogramme présenté ci-dessous (avec $\alpha = 7.38, \beta = 0.5$) :

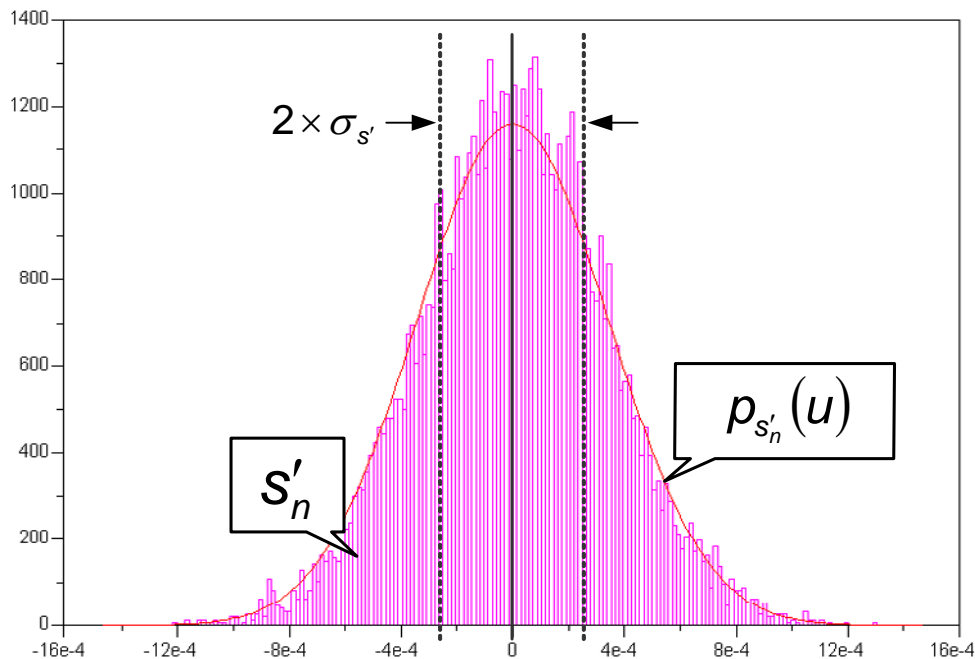


Figure 3 - 22 : Sortie du filtre s'_n

Nous approximons (après plusieurs itérations) cette répartition par la fonction suivante :

$$P_{s'_n}^*(u) = \frac{1}{2 \cdot \pi \cdot A \cdot G \cdot F_{ech} \cdot \sigma_{total} \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(u)^2}{2(2 \cdot \pi \cdot A \cdot F_{ech} \cdot G \cdot \sigma_{total})^2}} \text{ avec } G = 8.48$$

Eq. 3 - 30

III.2.3. Simulation d'un modèle MOS du filtre

Pour la simulation nous avons recours à plusieurs types de modélisation pour effectuer une simulation temporelle sous ADVanceMS [ADV00]. Pour coller au plus près de la réalité, nous avons repris les modèles Eldo [Eld04] fournis par le fondeur AMS [AMS] et laissons intacte la structure comportant les capacités, les interrupteurs et l'Amplificateur Opérationnel à Transconductance (niveau transistor). Par contre pour introduire un jitter en interne nous avons eu recours à une modélisation en langage VHDL-AMS [IEE99] du générateur d'horloges non-recouvrantes. Le modèle est explicité ci-dessous :

```

-----
ENTITY no_clocks_gene_ana IS
  GENERIC(delay_up      : REAL := 1.97e-9;
          delay_down    : REAL := 1.82e-9;
          delay1        : REAL := 0.5e-9;
          delay2        : REAL := 1.0e-9;
          sigma         : REAL := 0.0);
  PORT(  TERMINAL VSS : ELECTRICAL ;
        TERMINAL VDD : ELECTRICAL ;
        TERMINAL C26M : ELECTRICAL ;
        TERMINAL CK : ELECTRICAL_VECTOR (0 TO 7));
END ENTITY no_clocks_gene_ana;
-----
ARCHITECTURE behaviour OF no_clocks_gene_ana IS

CONSTANT Trans      : REAL := 10.0e-12;

QUANTITY Vplus      ACROSS VDD TO electrical_ground;
QUANTITY Vminus     ACROSS VSS TO electrical_ground;
QUANTITY V_C26M     ACROSS C26M TO electrical_ground;
QUANTITY V_CK_0     ACROSS I_CK_0 THROUGH CK(0) TO electrical_ground;
QUANTITY V_CK_1     ACROSS I_CK_1 THROUGH CK(1) TO electrical_ground;
QUANTITY V_CK_2     ACROSS I_CK_2 THROUGH CK(2) TO electrical_ground;
QUANTITY V_CK_3     ACROSS I_CK_3 THROUGH CK(3) TO electrical_ground;
QUANTITY V_CK_4     ACROSS I_CK_4 THROUGH CK(4) TO electrical_ground;
QUANTITY V_CK_5     ACROSS I_CK_5 THROUGH CK(5) TO electrical_ground;
QUANTITY V_CK_6     ACROSS I_CK_6 THROUGH CK(6) TO electrical_ground;
QUANTITY V_CK_7     ACROSS I_CK_7 THROUGH CK(7) TO electrical_ground;

SIGNAL clock_in      : BIT := '0';
SIGNAL latch_ADC     : BIT := '0';
SIGNAL clk_out       : BIT_VECTOR (0 TO 7) := ('1','0','0','1','1','0','0','1');
SIGNAL clk_out_real  : REAL_VECTOR (0 TO 7) :=
    (Vplus,Vminus,Vminus,Vplus,Vplus,Vminus,Vminus,Vplus);

BEGIN

clock_gene : PROCESS (V_C26M'ABOVE((Vplus-Vminus)/2.0))
BEGIN IF (V_C26M'ABOVE((Vplus-Vminus)/2.0)) = TRUE
    THEN clock_in <= '1'; ELSE clock_in <= '0'; END IF;
END PROCESS clock_gene;

clock : ENTITY no_clocks_gene(behaviour)
GENERIC MAP ( delay_up => delay_up, delay_down => delay_down, delay0
=> 0.0, delay1 => delay1, delay2 => delay2, sigma => sigma)
PORT MAP ( clock_in => clock_in , clock_out => clock_out,
latch_ADC => latch_ADC);

PROCESS (clock_out)
BEGIN FOR i IN 0 TO 7 LOOP
IF clock_out(i)='1' THEN clock_out_real(i)<= Vplus;
ELSE clock_out_real(i)<= Vminus; END IF;

```

DEFINITION des retards pour le diviseur de fréquence par deux puis pour les différentes horloges non-recouvrantes.

DEFINITION des CONNECTIONS d'entrée/sortie, Les tensions d'alimentations qui serviront à définir la tension de seuil, puis l'horloge de commande et enfin les horloges non-recouvrantes.

DEFINITION des QUANTITY en relation avec le TERMINAL défini précédemment.

Ce modèle est mixte (numérique/ analogique) les signaux d'entrées analogiques sont convertis en signaux numérique pour générer numériquement les signaux interne d'horloges. Ces signaux numériques sont ensuite convertis en signaux analogiques.

Nous utilisons ici un modèle qui est décrit dans le Chapitre IV. Il génère les horloges non-recouvrantes avec jitter.

Conversion des signaux numériques en signaux analogiques. Nous utilisons la fonction BREAK qui synchronise les noyaux de simulation numérique et


```

END LOOP; END PROCESS;

BREAK ON clock_out_real;
V_CK_0 == clock_out_real(0)'RAMP(trans,trans);
V_CK_1 == clock_out_real(1)'RAMP(trans,trans);
V_CK_2 == clock_out_real(2)'RAMP(trans,trans);
V_CK_3 == clock_out_real(3)'RAMP(trans,trans);
V_CK_4 == clock_out_real(4)'RAMP(trans,trans);
V_CK_5 == clock_out_real(5)'RAMP(trans,trans);
V_CK_6 == clock_out_real(6)'RAMP(trans,trans);
V_CK_7 == clock_out_real(7)'RAMP(trans,trans);
END ARCHITECTURE behaviour;

```

analogique

Code 3 - 1 : Modèle en VHDL-AMS du générateur d'horloges non-recouvrantes

Une série de simulations à diverses fréquences d'entrées comprise entre 500kHz et 3MHz a été réalisée. Après échantillonnage de la sortie, nous avons calculé l'écart type en utilisant l'Eq. 3 - 29. Le résultat de ces simulations au niveau des transistors est résumé dans le tableau ci-dessous :

Fréquence (MHz)	0.5e6	1.0e6	1.5e6	2.0e6	2,50e6	3e6
G	8,209	8,205	8,203	8,191	8,204	8,220

Tab 3 - 3

Nous trouvons alors un facteur G légèrement différent de celui trouvé précédemment, nous prendrons $G = 8.2$ qui sont semble le plus proche de la réalité au regard des conditions d'obtention.

La figure ci-dessous reprend les histogrammes que nous avons réalisés :

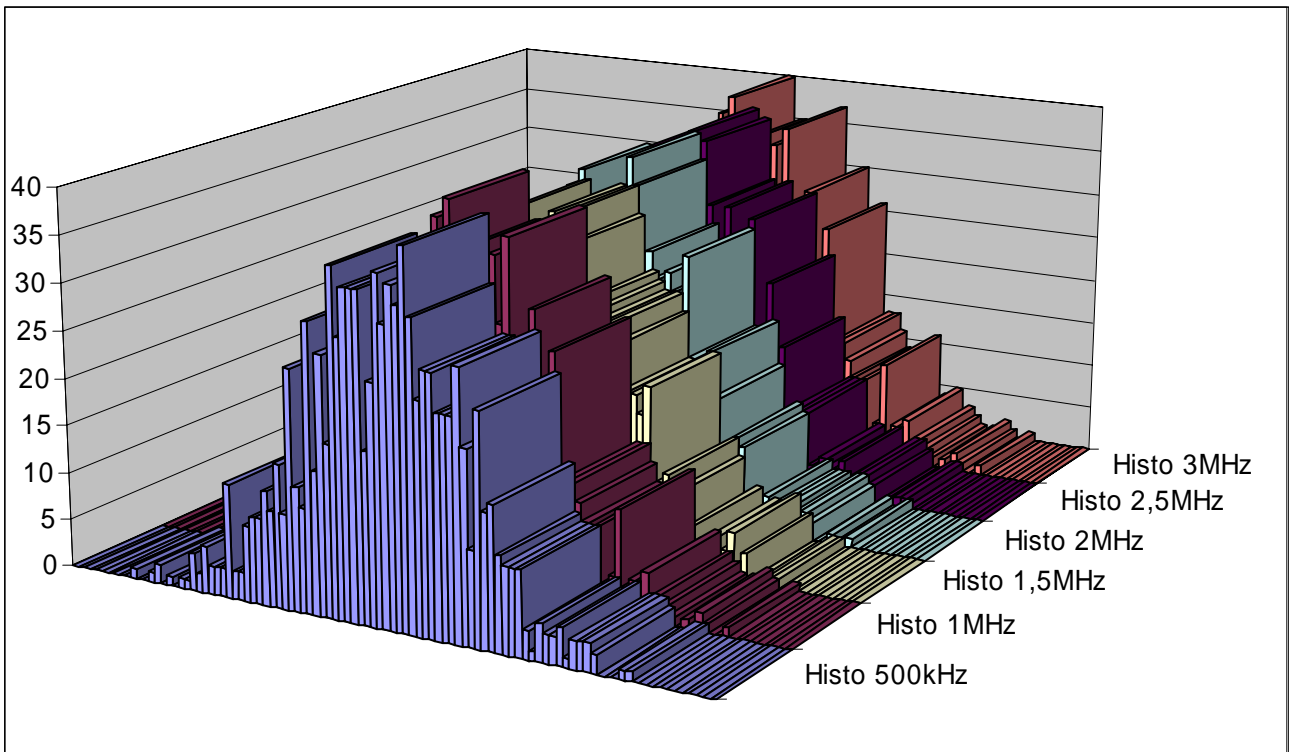


Figure 3 - 23 : Histogrammes du filtre S'_n pour différentes fréquences d'entrée

IV. ANNEXES : Chapitre IV

IV.1. Modèle de l'amplificateur à Contrôle Automatique de Gain (CAG)

Nous présentons ci-dessous la modélisation du noyau fonctionnel de l'Amplificateur AGC (Automatic Gain Control). Afin de pouvoir s'interfacer avec le modèle transistor des connexions telles que *VDD*, *VSS* ont été ajoutées mais ne sont utilisées que dans les niveaux supérieurs (*Full*). Le modèle de l'amplificateur CAG ne présente pas une grande complexité car reprend les équations électriques de chaque nœud :

```
-----
ENTITY AGC IS
PORT ( TERMINAL VDD, VSS, MC      : ELECTRICAL;
      TERMINAL I100N              : ELECTRICAL;
      TERMINAL RFP, RFN           : ELECTRICAL;
      TERMINAL OUTP, OUTN         : ELECTRICAL;
      SIGNAL CMD_AGC              : IN BIT_VECTOR (2 DOWNTO 0);
      SIGNAL GAINMAX_ADJ          : IN BIT;
      SIGNAL IADJ                 : IN BIT_VECTOR (1 DOWNTO 0));
END ENTITY AGC;
```

```
-----
ARCHITECTURE behaviour_kernel OF AGC IS;
```

```
CONSTANT Covl      : REAL := 0.7e-12;
CONSTANT Cin_OTA   : REAL := 2.0e-12;
CONSTANT GMo       : REAL := 0.04175;
```

```
TERMINAL OTA_p, OTA_n      : ELECTRICAL;
QUANTITY Vota              ACROSS          OTA_p      TO OTA_n;
QUANTITY Votap             ACROSS lotap   THROUGH OTA_p TO electrical_ground;
QUANTITY Votan             ACROSS lotan   THROUGH OTA_n TO electrical_ground;
QUANTITY V_bias            ACROSS I_bias  THROUGH I100N TO VSS;
QUANTITY Vin               ACROSS lin     THROUGH RFP  TO RFN;
QUANTITY Vrin_p            ACROSS Irin_p  THROUGH RFP  TO OTA_p;
QUANTITY Vrin_n            ACROSS Irin_n  THROUGH RFN  TO OTA_n;
QUANTITY Vrfb_p            ACROSS lrfb_p , lcovl_p THROUGH OTA_p TO OUTP;
QUANTITY Vrfb_n            ACROSS lrfb_n , lcovl_n THROUGH OTA_n TO OUTN;
QUANTITY Vout              ACROSS lout    THROUGH OUTP TO OUTN;
```

```
SIGNAL Rze      : REAL := 1.0e+12;
SIGNAL Rfb      : REAL := 462.3;
SIGNAL Rin      : REAL := 1.0;
SIGNAL RI       : REAL := 1.0e+6;
SIGNAL Gain     : INTEGER := 0;
SIGNAL en       : REAL := 0.0;
```

```
BEGIN
```

```
--Reglage du gain
```

-- G2	G1	G0	Gain	Rze	Rfb	Rin	en	Gain en dB
-- 0	0	0	0	1.0e+12	462.3	1.0	0.0	OFF
-- 0	0	1	1	1.0e+12	462.3	1.0	1.0	Gain max = 21dB
-- 0	1	0	2	192.0	426.3	36.0	1.0	Gain max-6dB = 15dB
-- 0	1	1	3	124.0	370.5	91.8	1.0	Gain max-12dB = 9dB
-- 1	0	0	4	105.3	296.1	166.2	1.0	Gain max-18dB = 3dB
-- 1	0	1	5	98.0	212.1	250.2	1.0	Gain max-24dB = -3dB
-- 1	1	0	6	94.6	137.7	324.6	1.0	Gain max-30dB = -9dB
-- 1	1	1	-1	94.6	137.7	324.6	1.0	Gain N.D. idem gain=6

*DEFINITION DES
CONNECTIONS I/O :*
*alimentation en
tension et courant*
- entrée différentielle
- sortie différentielle
- commande du gain
*- commande de la
consommation*

*DEFINITION DES
CONSTANTES :*
capacités, gain
*transconductance de
l'AOT*

*DEFINITION DU COUPLE
TENSION / COURANT
POUR LES NŒUDS
INTERNES ET
EXTERNNES*

*DEFINITION DES
RESISTANCES SOUS
FORME DE SIGNAUX :*
Rin=x.Rfb et Rfb=(1-x).Rfb

```

PROCESS(GAINMAX_ADJ,CMD_AGC)
CONSTANT G_ze      : REAL_VECTOR(2 TO 6) := ((1.0/192.0),(1.0/350.0),
                                           (1.0/700.0),(1.0/1.4e+3),(1.0/2.8e+3));
CONSTANT Reosta    : REAL_VECTOR(2 TO 6) := (36.0, 55.8, 74.4, 84.0, 74.4);
VARIABLE i         : INTEGER := 0;
VARIABLE G_ain     : INTEGER := 0;
VARIABLE Gze       : REAL := 1.0/1.0e+12;
VARIABLE R_fb      : REAL := 462.3;

BEGIN

IF CMD_AGC(0) = '1' THEN G_ain := G_ain + 1; END IF;
IF CMD_AGC(1) = '1' THEN G_ain := G_ain + 2; END IF;
IF CMD_AGC(2) = '1' THEN G_ain := G_ain + 4; END IF;

FOR i IN 2 TO 6 LOOP
IF (i <= G_ain) THEN Gze := Gze + G_ze(i); END IF;
IF (i <= G_ain) THEN R_fb := R_fb - Reosta(i); END IF;
END LOOP;

Rze <= 1.0/Gze;
Rfb <= R_fb;
IF (462.3-R_fb) > 0.0 THEN Rin <= 462.3-R_fb; ELSE Rin <= 1.0; END IF;
IF G_ain /= 7 THEN Gain <= G_ain; ELSE Gain <= -1; END IF;
IF G_ain /= 0 THEN en <= 1.0; END IF;
IF (GAINMAX_ADJ = '1') AND (G_ain=1) THEN RI <= 6.0e+3;ELSE RI <= 1.0e+6; END IF;
END PROCESS;

lotap == Cin_ota*Votap'DOT;
lotan == Cin_ota*Votan'DOT;

V_bias == I_bias*1.0e+6;

Vin == Iin*Rze;
Vrin_p == Irin_p*Rin;
Vrin_n == Irin_n*Rin;

Vrfb_p == Irfb_p*Rfb;
Icovl_p == Covl*Vrfb_p'DOT;
Vrfb_n == Irfb_n*Rfb;
Icovl_n == Covl*Vrfb_n'DOT;

Iout == en*GMo*Vota + Vout/ri;

END ARCHITECTURE behaviour_kernel;

```

CALCUL DES
VALEURS DES
RESISTANCES,
FONCTION DES
BITS DE COMMANDE

MISE EN EQUATIONS
- capacité d'entrée de
l'AOT

- courant de
consommation
- résistance Rin

-résistance Rfb

-courant de sortie de
l'AOT

Code 4 - 2 : Modèle en VHDL-AMS de l'amplificateur CAG

IV.2. Modèle du générateur de quadrature

Comme les signaux d'entrée / sorties sont supposés idéaux, les horloges sont de classe *BIT*, leur valeur sera donc '0' ou '1' sans état intermédiaire, ni temps de transition entre deux états.

```

ENTITY quadrature IS
    PORT ( SIGNAL clock_in      : IN BIT;
          SIGNAL clock_out     : INOUT BIT_VECTOR (0 TO 3) := ('0','1','0','1'));
END ENTITY quadrature ;

ARCHITECTURE behaviour OF quadrature IS

```

DEFINITION DES
CONNECTIONS I/O :
- horloge entrant / sortant
(classe : BIT)

```

BEGIN
  Output_generation : PROCESS (clock_in)
  BEGIN
  IF clock_in = '1' THEN
    IF clock_out = "0101" THEN clock_out <= "1001";END IF;
    IF clock_out = "1001" THEN clock_out <= "1010";END IF;
    IF clock_out = "1010" THEN clock_out <= "0110";END IF;
    IF clock_out = "0110" THEN clock_out <= "0101";END IF;
  END IF;
  END PROCESS Output_generation;
END ARCHITECTURE behaviour;

```

Déclenchement sur front montant de l'horloge entrant (PROCESS)

Table de vérité et génération des horloges de sortie

Code 4 - 3 : Modèle en VHDL-AMS du générateur de quadrature

N.B. : des retards, ou des temps de montée / descente pourront éventuellement être ajoutés lors d'une conversion *BIT / QUANTITY* .

IV.3. Modèle de Mélangeur

Nous avons choisi de subdiviser ce modèle en 2 : une partie modélise un interrupteur et l'autre définit la valeur de la résistance en fonction des bits de commande. Mis à part cela, le modèle reprend le schéma fonctionnel représenté par la Figure 4 - 9.

```

-----
ENTITY switch IS
  GENERIC ( t_rise,t_fall : REAL := 100.0e-12);
  PORT ( TERMINAL Psw_I, Psw_O : ELECTRICAL;
        SIGNAL Rsw : IN REAL);
        SIGNAL CMD : IN BIT);
END ENTITY switch;
-----
ARCHITECTURE behaviour OF switch IS

SIGNAL Gsw : REAL := 1.0;
QUANTITY Vsw ACROSS Isw THROUGH Psw_I TO Psw_O;

BEGIN
BREAK ON CMD;
PROCESS (CMD)
BEGIN
  IF CMD = '1' THEN Gsw <= 1.0/(Ron+Rsw); ELSE Gsw <= 0.0; END IF;
END PROCESS;
Isw == Vsw*Gsw*RAMP(t_rise,t_fall);
END ARCHITECTURE behaviour;
-----
PACKAGE mixer_function IS
FUNCTION R (code: BIT_VECTOR(3 DOWNTO 0)) RETURN REAL;
END PACKAGE mixer_function;

PACKAGE BODY mixer_function IS
FUNCTION R (code: BIT_VECTOR(3 DOWNTO 0)) RETURN REAL IS
  VARIABLE res : REAL;
  VARIABLE G : REAL := 1.0/2.2e+3;
BEGIN
  IF code(3)='0' THEN G := G+1.0/2.431e+3;END IF;
  IF code(2)='0' THEN G := G+1.0/4.862e+3;END IF;
  IF code(1)='0' THEN G := G+1.0/9.724e+3;END IF;

```

DEFINITION DE L'INTERRUPTEUR ELEMENTAIRE QUI COMPOSE LE MIXER.

DEFINITION DES CONNECTIONS I/O :
- entrée sortie
- commande

DEFINITION DU COUPLE TENSION / COURANT

CALCUL DE LA VALEURS DE LA RESISTANCE, FONCTION DU BITS DE COMMANDE.

DEFINITION D'UNE FONCTION POUR LA RESISTANCE VARIABLE Rmix

```

IF code(0)='0' THEN G := G+1.0/19.448e+3;END IF;
res := 1.0/G;
RETURN (res);
END FUNCTION;
END PACKAGE BODY mixer_function;
-----
ENTITY mixer IS
    PORT ( TERMINAL VDD, VSS      : ELECTRICAL;
          TERMINAL Pin_p, Pin_n  : ELECTRICAL;
          TERMINAL Pout_p, Pout_n : ELECTRICAL;
          SIGNAL clock_in        : IN BIT_VECTOR(0 TO 1);
          SIGNAL CMD_MIX         : IN BIT_VECTOR (3 DOWNT0 0):="0111");
END ENTITY mixer;
-----
ARCHITECTURE behaviour_kernel OF mixer IS

CONSTANT t_rise      : REAL := 100.0e-12;
CONSTANT t_fall     : REAL := 100.0e-12;

BEGIN

PROCESS (CMD_MIX)
BEGIN Rsw <= R(CMD_MIX);
END PROCESS;

switchp : ENTITY switch (behaviour)
GENERIC MAP ( t_rise=> t_rise, t_fall=> t_fall)
PORT MAP (    Psw_I=>Pin_p, Psw_O=>Pout_p, Rsw=>Rsw,CMD=>clock_in(0));

switchn : ENTITY switch_eldo(behaviour_dig)
GENERIC MAP ( t_rise=> t_rise, t_fall=> t_fall)
PORT MAP (    Psw_I=>Pin_n, Psw_O=>Pout_n, Rsw=>Rsw,CMD=>clock_in(0));

switch_invp : ENTITY switch_eldo(behaviour_dig)
GENERIC MAP ( t_rise=> t_rise, t_fall=> t_fall)
PORT MAP (    Psw_I=>Pin_p, Psw_O=>Pout_n, Rsw=>Rsw,CMD=>clock_in(1));

switch_invn : ENTITY switch_eldo(behaviour_dig)
GENERIC MAP ( t_rise=> t_rise, t_fall=> t_fall)
PORT MAP (    Psw_I=>Pin_n, Psw_O=>Pout_p, Rsw=>Rsw,CMD=>clock_in(1));

END ARCHITECTURE behaviour_kernel;
-----

```

*DEFINITION DES
CONNECTIONS I/O :*
- alimentation en
tension
- entrées / sorties
différentielles
- commande des
interrupteurs
- commande de la valeur
de Rmix

*UTILISATION DU SOUS-
CIRCUIT switch,
PRECEDEMMENT
DEFINIT.*

Code 4 - 4 : Modèle en VHDL-AMS du Mélangeur

IV.4. Modèle du filtre Anti-repliement (Anti Aliasing Filter)

Comme l'architecture du bloc n'est pas encore protégée par un brevet, nous ne pouvons développer les équations électriques qui constituent la fonction de transfert du filtre.

```

-----
ENTITY AAF_complex IS
PORT ( TERMINAL VDD, VSS, MC      : ELECTRICAL;
       TERMINAL I20u              : ELECTRICAL;
       TERMINAL Pin_lp, Pin_ln    : ELECTRICAL;
       TERMINAL Pin_qp, Pin_qn    : ELECTRICAL;
       TERMINAL Pout_lp, Pout_ln   : ELECTRICAL;
       TERMINAL Pout_qp, Pout_qn   : ELECTRICAL;
       SIGNAL EN                   : BIT := '1';
       SIGNAL CMD_AAF              : BIT_VECTOR (2 DOWNT0 0) := "011";
       SIGNAL IADJ                 : BIT_VECTOR (1 DOWNT0 0) := "01");

```

*DEFINITION DES
CONNECTIONS I/O :*
- alimentation en
tension et courant
- entrée différentielle I et Q
- sortie différentielle I et Q
- enable
- commande des
résistances
- commande de la

```
END ENTITY AAF_complex;
```

```
-----
ARCHITECTURE behaviour_kernel OF AAF_complex IS
```

```
CONSTANT Ra      : REAL := 1.0;
CONSTANT C_1     : REAL := 18.3e-12;
CONSTANT C_2     : REAL := 34.8e-12;
CONSTANT gm      : REAL := 10.0;
CONSTANT Rin     : REAL := 1.0;

-- Input
TERMINAL Pin_I, Pin_Q      : ELECTRICAL;
TERMINAL Pout_I, Pout_Q    : ELECTRICAL;
QUANTITY Vin_I_diff ACROSS Pin_Ip      TO Pin_In;
QUANTITY Vin_Q_diff ACROSS Pin_Qp      TO Pin_Qn;
QUANTITY Vin_I ACROSS lin_I THROUGH Pin_I TO electrical_ground;
QUANTITY Vin_Q ACROSS lin_Q THROUGH Pin_Q TO electrical_ground;
QUANTITY Vin_Ip ACROSS lin_Ip THROUGH Pin_Ip TO MC;
QUANTITY Vin_In ACROSS lin_In THROUGH Pin_In TO MC;
QUANTITY Vin_Qp ACROSS lin_Qp THROUGH Pin_Qp TO MC;
QUANTITY Vin_Qn ACROSS lin_Qn THROUGH Pin_Qn TO MC;

-- Voie I
TERMINAL PA_I, PO_I, PB_I : ELECTRICAL;
QUANTITY VA_I ACROSS IA_I THROUGH PA_I TO electrical_ground;
QUANTITY VO_I ACROSS IO_I THROUGH PO_I TO electrical_ground;
QUANTITY VB_I ACROSS IB_I THROUGH PB_I TO electrical_ground;
QUANTITY Vout_I ACROSS Iout_I THROUGH Pout_I TO electrical_ground;

-- Voie Q
TERMINAL PA_Q, PO_Q, PB_Q : ELECTRICAL;
QUANTITY VA_Q ACROSS IA_Q THROUGH PA_Q TO electrical_ground;
QUANTITY VO_Q ACROSS IO_Q THROUGH PO_Q TO electrical_ground;
QUANTITY VB_Q ACROSS IB_Q THROUGH PB_Q TO electrical_ground;
QUANTITY Vout_Q ACROSS Iout_Q THROUGH Pout_Q TO electrical_ground;

-- Output
QUANTITY Vout_Ip ACROSS Iout_Ip THROUGH Pout_Ip TO electrical_ground;
QUANTITY Vout_In ACROSS Iout_In THROUGH Pout_In TO electrical_ground;
QUANTITY Vout_Qp ACROSS Iout_Qp THROUGH Pout_Qp TO electrical_ground;
QUANTITY Vout_Qn ACROSS Iout_Qn THROUGH Pout_Qn TO electrical_ground;
QUANTITY V_bias ACROSS I_bias THROUGH I20u TO VSS;
```

```
SIGNAL Rb      : REAL := 5680.0;
SIGNAL R_2     : REAL := 3550.0;
SIGNAL R_3     : REAL := 7100.0;
SIGNAL Rc_1    : REAL := 5339.2;
SIGNAL Rc_2    : REAL := 2840.0;
```

```
BEGIN
```

```
PROCESS (CMD_AAF)
VARIABLE res : REAL_VECTOR(0 TO 4) := (5680.0,3550.0,7100.0,5339.2,2840.0);
BEGIN
res := R(CMD_AAF);
Rb <= res(0); R_2 <= res(1); R_3 <= res(2); Rc_1 <= res(3); Rc_2 <= res(4);
END PROCESS;
```

```
V_bias == I_bias*1.0e+6;
```

```
-- Input
```

```
Vin_I_diff == Vin_I;
Vin_Q_diff == Vin_Q;
Vin_Ip == Rin*lin_Ip;
Vin_In == Rin*lin_In;
Vin_Qp == Rin*lin_Qp;
Vin_Qn == Rin*lin_Qn;
```

```
-- Voie I
```

```
XXXXXXXXXXXX Equation de la voie I XXXXXXXXXXXXXXXXXXXX
```

```
-- Voie Q
```

```
XXXXXXXXXXXX Equation de la voie Q XXXXXXXXXXXXXXXXXXXX
```

```
-- Output
```

consommation

DEFINITION DES
CONSTANTES :
capacités, gain
transconductance des
AOT

DEFINITION DU COUPLE
TENSION / COURANT
POUR LES NŒUDS
INTERNES ET
EXTERNNES

DEFINITION DES
RESISTANCES SOUS
FORME DE SIGNAUX.

CALCUL DES
VALEURS DES
RESISTANCES,
FONCTION DES
BITS DE COMMANDE

MISE EN EQUATIONS
-courant de consommation
-passage d'entrée
différentielle à mono-voie
-impédances d'entrée du
montage

-équations voie I

-équations voie Q

```

Vout_Ip      == Vout_I/2.0;
Vout_In      == -Vout_I/2.0;
Vout_Qp      == Vout_Q/2.0;
Vout_Qn      == -Vout_Q/2.0;
END ARCHITECTURE behaviour_kernel;

```

-passage de la sortie de
mono_voie à différentielle

Code 4 - 5 : Modèle en VHDL-AMS du filtre anti-repliements

IV.5. Modèle de la PLL

Ce modèle est basé sur le modèle d'un générateur d'horloge avec jitter présenté dans le Chapitre II (cf. §IV.2). Les principales différences proviennent du calcul en interne de la période d'entrée et de l'utilisation d'une fonction d'ajout de bruit spécifique au bruit de la PLL.

IV.5.1. Fonction du bruit de phase de la PLL

```

FUNCTION PLL_noise (Fsmpl, PSD_dBv_Hz, fc_f1, fc_f2 :REAL;
                   int: REAL_VECTOR(0 TO 17)) RETURN REAL_VECTOR IS

  CONSTANT Tsmpl      : REAL := 1.0/Fsmpl;
  CONSTANT tau        : REAL := 1.0/(MATH_2_PI*fc_f2);
  CONSTANT option     : REAL_VECTOR(2 DOWNT0 0) := (1.0,1.0,0.0);
  CONSTANT Beta_2     : INTEGER := 2;
  VARIABLE outt      : REAL_VECTOR (0 TO 17);

  BEGIN
    outt(1 TO 17):= white_flicker_noise (Fsmpl, PSD_dBv_Hz, fc_f1, Beta_2, option, int(1 TO 17));
    outt(0) := (1.0/(Tsmpl+2.0*tau))*(Tsmpl*(outt(1)+int(1))-(Tsmpl-2.0*tau)*int(0));
  RETURN (outt);

END FUNCTION;

```

Cette modélisation est basée sur la fonction de génération de bruit blanc et de flicker définie dans le Chapitre II.

Un filtre passe bas est également appliqué pour former une pente à -20dB/décade

Code 4 - 6 : Modèle du bruit de phase de la PLL

IV.5.2. Modèle de la PLL

```

-----
ENTITY PLL IS
  GENERIC (
    PSD_dBc_Hz_white : REAL := -125.0;
    fc_flicker       : REAL := 5.5e+3;
    fc_f_2           : REAL := 250.0e+3;
  )
  PORT (
    SIGNAL Nref      : IN BIT := '0';
    SIGNAL Npulse    : INOUT BIT := '0';
  );
END ENTITY PLL;
-----
ARCHITECTURE behaviour OF PLL IS

  CONSTANT N_divider : REAL := 11.0;
  SIGNAL period      : REAL := 0.0;

  BEGIN
    period_determination : PROCESS
      VARIABLE tbegin    : REAL := 0.0;
    BEGIN

```

DEFINITION DES
PARAMETRES :
puissance du bruit,
fréquences de coupure.
DEFINITION DES
CONNECTIONS I/O :
- horloge de réf
- horloge sortie

Définition du facteur de
division.

PROCESS:

détermination de la

```

        IF tbegin = 0.0 THEN tbegin := NOW;
        ELSE period <= (NOW - tbegin)/N_divider; tbegin := NOW; END IF;
WAIT UNTIL Nref = '1';
END PROCESS period_determination;

Output_generation : PROCESS
VARIABLE period_int      : REAL := 0.0;
VARIABLE Phi_n, Phi_n_1  : REAL := 0.0;
VARIABLE rand_ph         : REAL_VECTOR (0 TO 17) := (i.c.);

BEGIN
IF period = 0.0 THEN WAIT UNTIL period > 0.0;
ELSE
    Phi_n_1 := Phi_n;
    rand_ph:=PLL_noise(1.0/period,PSD_dBc_Hz_white,fc_flicker,fc_f_2,rand_ph);
    Phi_n := rand_ph(0);

    IF ((1.0 + ((Phi_n-Phi_n_1)/MATH_2_PI))*period > 0.0) THEN
    period_int := (1.0 + ((Phi_n-Phi_n_1)/MATH_2_PI))*period; END IF;
    Npulse <= '1'; WAIT FOR 0.5*period_int;
    Npulse <= '0'; WAIT FOR 0.5*period_int;
END IF;
END PROCESS Output_generation;

END ARCHITECTURE behaviour;

```

période du signal entrant

génération de l'aléa
de la phase

front montant
front descendant

Code 4 - 7 : Modèle en VHDL-AMS de la PLL

IV.6. Modèle de la chaîne FI

```

-----
ENTITY FI IS
    GENERIC (Tsmp      : REAL := 1.0e-6);
    PORT ( TERMINAL VDD, VSS, MC      : ELECTRICAL;
          TERMINAL I20u, I100N      : ELECTRICAL;
          TERMINAL Pinp, Pinn       : ELECTRICAL;
          TERMINAL Pout_lp, Pout_In  : ELECTRICAL;
          TERMINAL Pout_Qp, Pout_Qn  : ELECTRICAL;
          SIGNAL C26M                : BIT:= '1';
          SIGNAL CMD_AGC              : BIT_VECTOR(2 DOWNTO 0):="001";
          SIGNAL IADJ_AGC             : BIT_VECTOR(1 DOWNTO 0):="11";
          SIGNAL GAINMAX_ADJ         : BIT := '1';
          SIGNAL CMD_MIX              : BIT_VECTOR(3 DOWNTO 0):="1000";
          SIGNAL CMD_AAF              : BIT_VECTOR(2 DOWNTO 0):="000";
          SIGNAL EN_AAF              : BIT:= '1';
          SIGNAL IADJ_AAF            : BIT_VECTOR(1 DOWNTO 0):="00");
END ENTITY FI;

-----
ARCHITECTURE behaviour OF FI IS

    CONSTANT AGC_noise : REAL := 0.9e-9;      -- Vrms_sqrt_Hz
    CONSTANT MIXER_noise: REAL := 0.0;        -- Vrms_sqrt_Hz
    CONSTANT AAF_noise  : REAL := 15.0e-9;    -- Vrms_sqrt_Hz

    SIGNAL clock_PLL    : BIT := '0';
    SIGNAL clock_mixer  : BIT_VECTOR(0 TO 3);

    TERMINAL Pout_AGC_p, Pout_AGC_n : ELECTRICAL;
    TERMINAL Pout_MIX_lp, Pout_MIX_In : ELECTRICAL;
    TERMINAL Pout_MIX_Qp, Pout_MIX_Qn : ELECTRICAL;

BEGIN

    AGC_amp : ENTITY AGC (behaviour)

```

DEFINITION DES
CONNECTIONS I/O :
- alimentations
- courants de pol°
- entrée diff
- horloge entrant
- codes sortants voie I et
Q
- enable

DEFINITION DU COUPLE
TENSION / COURANT
POUR LES NŒUDS
INTERNES ET
EXTERNNES

AGC


```

GENERIC MAP ( noise_Vrms_sqrt_Hz=>AGC_noise, Tsmpl=>Tsmpl)
PORT MAP ( VDD=>VDD, VSS=>VSS, MC=>MC, I100N=>I100N,
RFP=>Pinp, RFN=>Pinn,OUTP=>Pout_AGC_p,
OUTN=>Pout_AGC_n, CMD_AGC=>CMD_AGC,
GAINMAX_ADJ=>GAINMAX_ADJ, IADJ=>IADJ_AGC);

```

```

PLL_RX : ENTITY PLL (behaviour)
GENERIC MAP ( PSD_dBc_Hz_white=>-125.0, fc_flicker=>5.5e+3,
fc_f_2=>250.0e+3)
PORT MAP ( Nref => C26M, Npulse => clock_PLL);

```

```

quadra : ENTITY quadrature (behaviour)
PORT MAP ( clock_in=>clock_PLL, clock_out=>clock_mixer);

```

```

mixe_I : ENTITY mixer (behaviour)
GENERIC MAP ( noise_Vrms_sqrt_Hz=>MIXER_noise, Tsmpl=>Tsmpl)
PORT MAP ( VDD=>VDD, VSS=>VSS, Pin_p=>Pout_AGC_p,
Pin_n=>Pout_AGC_n, Pout_p=>Pout_MIX_Ip,
Pout_n=>Pout_MIX_In, CMD_MIX=>CMD_MIX,
clock_in(0)=>clock_mixer(2), clock_in(1)=>clock_mixer(3));

```

```

mixe_Q : ENTITY mixer (behaviour)
GENERIC MAP ( noise_Vrms_sqrt_Hz=>MIXER_noise, Tsmpl=>Tsmpl)
PORT MAP ( VDD=>VDD, VSS=>VSS, Pin_p=>Pout_AGC_p,
Pin_n=>Pout_AGC_n, Pout_p=>Pout_MIX_Qp,
Pout_n=>Pout_MIX_Qn, CMD_MIX=>CMD_MIX,
clock_in(0)=>clock_mixer(0), clock_in(1)=>clock_mixer(1));

```

```

AAF_filtering : ENTITY AAF_complex (behaviour)
GENERIC MAP ( noise_Vrms_sqrt_Hz=>AAF_noise, Tsmpl=>Tsmpl)
PORT MAP ( VDD=>VDD, VSS=>VSS, MC=>MC, I20u=>I20u,
Pin_Ip=>Pout_MIX_Ip, Pin_In=>Pout_MIX_In,
Pin_Qp=>Pout_MIX_Qp, Pin_Qn=>Pout_MIX_Qn,
Pout_Ip=>Pout_Ip, Pout_In=>Pout_In,
Pout_Qp=>Pout_Qp, Pout_Qn=>Pout_Qn,
EN=>EN_AAF,CMD_AAF=>CMD_AAF,IADJ=>IADJ_AAF);

```

END ARCHITECTURE behaviour;

PLL à 286MHz

GENERATEUR de quadrature

MIXER voie I

MIXER voie Q

Filtre Anti-Repliements

Code 4 - 8 : Modèle en VHDL-AMS de la chaîne FI

IV.7. Modèle du générateur d'horloges non-recouvrantes

```

LIBRARY LIBRAIRIE_BRUIT;
USE LIBRAIRIE_BRUIT.fonction_random.all;

ENTITY no_clocks_gene IS
GENERIC ( delay_up : REAL := 1.97e-9;
delay_down : REAL := 1.82e-9;
delay_0 : REAL := 1.0e-9;
delay_1 : REAL := 0.5e-9;
delay_2 : REAL := 1.0e-9;
sigma : REAL := 0.0;
ini_cond : REAL_VECTOR (0 TO 4) := (0.0, 21483562.0,214.0,546.0,115.0));
PORT ( SIGNAL clock_in : IN BIT;
SIGNAL clock_out : INOUT BIT_VECTOR (0 TO 7) := ('1','0','0','1','1','0','0','1');
SIGNAL latch_ADC : INOUT BIT := '0');
END ENTITY no_clocks_gene;

```

ARCHITECTURE behaviour OF no_clocks_gene IS

```

CONSTANT d_0 : TIME := INTEGER(delay_0*1.0e+15)*fs;
CONSTANT d_1 : TIME := INTEGER(delay_1*1.0e+15)*fs;

```

DEFINITION DES PARAMETRES :
délais de non recouvrement

DEFINITION DES CONNECTIONS I/O :
- horloge entrant / sortant (BIT)

Conversion :
REAL -> TIME

```

CONSTANT d_2 : TIME := INTEGER(delay_2*1.0e+15)*fs;

BEGIN
Output_generation : PROCESS (clock_in)
  VARIABLE S : BIT := '0';
  VARIABLE rand : REAL_VECTOR (0 TO 4) := ini_cond;
  VARIABLE delay : TIME := INTEGER(delay_up*1.0e+15)*fs;
BEGIN
  IF clock_in = '1' THEN
    rand := gauss(sigma,rand);
    IF (S = '1') THEN
      delay := INTEGER((delay_down+rand(0))*1.0e+15)*fs;
      clock_out(2) <= NOT clock_out(2) AFTER delay;
      clock_out(3) <= NOT clock_out(3) AFTER delay;
      clock_out(6) <= NOT clock_out(6) AFTER delay+d_1;
      clock_out(7) <= NOT clock_out(7) AFTER delay+d_1;
      clock_out(4) <= NOT clock_out(4) AFTER delay+d_1+d_2;
      clock_out(5) <= NOT clock_out(5) AFTER delay+d_1+d_2;
      clock_out(0) <= NOT clock_out(0) AFTER delay+2.0*d_1+d_2;
      clock_out(1) <= NOT clock_out(1) AFTER delay+2.0*d_1+d_2;
      S := NOT S;
    ELSE
      delay := INTEGER((delay_up+rand(0))*1.0e+15)*fs;
      clock_out(0) <= NOT clock_out(0) AFTER delay;
      clock_out(1) <= NOT clock_out(1) AFTER delay;
      clock_out(4) <= NOT clock_out(4) AFTER delay+d_1;
      clock_out(5) <= NOT clock_out(5) AFTER delay+d_1;
      clock_out(6) <= NOT clock_out(6) AFTER delay+d_1+d_2;
      clock_out(7) <= NOT clock_out(7) AFTER delay+d_1+d_2;
      clock_out(3) <= NOT clock_out(3) AFTER delay+2.0*d_1+d_2;
      clock_out(2) <= NOT clock_out(2) AFTER delay+2.0*d_1+d_2;
      latch_ADC <= '1';
      S := NOT S;
    END IF;
  ELSE IF (S = '0') THEN
    latch_ADC <= '0' AFTER d_0;
  END IF;
END IF;
END PROCESS Output_generation;

END ARCHITECTURE behaviour;

```

Génération d'une horloge de fréquence/2, avec ajout de retard et du jitter

Cette nouvelle horloge interne sert ensuite de référentiel pour le déclenchement des PROCESS

Table de vérité pour les horloges de sorties non-recouvrantes

Génération du latch de déclenchement du CAN

Code 4 - 9 : Modèle en VHDL-AMS du générateur d'horloges non recouvrantes

IV.8. Modèle du Convertisseur Analogique/Numérique

IV.8.1. Le quantificateur

La modélisation ne présente pas de grande difficulté puisque nous nous intéressons au comportement idéal du Quantificateur.

```

ENTITY N_bits_Quantifier IS
  GENERIC (
    step : REAL := 46.38e-3;
    N_bits : INTEGER := 4;
    delay : REAL := 3.0e-9);
  PORT (
    SIGNAL latch : IN BIT;
    SIGNAL Ninp, Ninn : IN REAL;
    SIGNAL thermo_code : OUT BIT_VECTOR(0 TO (2**N_bits-2));
END ENTITY N_bits_Quantifier;

```

DEFINITION DES PARAMETRES :
-pas
-nombre de bits

DEFINITION DES CONNECTIONS I/O :
-latch
-entrée différentielle

```

-----
ARCHITECTURE behaviour OF N_bits_Quantifier IS
BEGIN
  codage : PROCESS (latch)
  VARIABLE code : BIT_VECTOR(0 TO (2**N_bits)-1) := (OTHERS => '0');
  BEGIN
  IF latch = '0' THEN
    FOR i IN 0 TO ((2**N_bits)-1) LOOP
      EXIT WHEN ((Ninp-Ninn) <= ((-(2.0**(N_bits-1) - 0.0) + REAL(i))*step));
      IF i /= 0 THEN code(i-1) := '1'; END IF;
    END LOOP;
    thermo_code <= code AFTER (delay*1.0e+12)*1ps;
    code := (OTHERS => '0');
  END IF;
  END PROCESS codage;
END ARCHITECTURE behaviour;
-----

```

-code thermomètre

PROCESS&BOUCLE :

Sur front descendant
calcul de la valeur du code
thermométrique.

Code 4 - 10 : Modèle en VHDL-AMS du Quantificateur

IV.8.2. L'encodeur logique

```

-----
ENTITY N_bits_Binary_Coder IS
  GENERIC ( N_bits : INTEGER := 4);
  PORT ( SIGNAL thermo_code : IN BIT_VECTOR(0 TO (2**N_bits-2));
        SIGNAL binary_code : OUT BIT_VECTOR(0 TO N_bits-1));
END ENTITY N_bits_Binary_Coder;
-----
ARCHITECTURE behaviour OF N_bits_Binary_Coder IS
BEGIN
  coder : PROCESS (thermo_code)
  VARIABLE counter : REAL := 0.0;
  BEGIN
  FOR i IN 0 TO 2**N_bits-2 LOOP
    IF thermo_code(i)='1' THEN counter := counter+1.0; END IF;
  END LOOP;

  FOR i IN N_bits-1 DOWNTO 0 LOOP
    IF (((counter / 2.0**i)-(counter MOD 2.0**i)/2.0**i) = 1.0)
    THEN binary_code(i)<='1';
    ELSE binary_code(i)<='0';
    END IF;
    counter := counter MOD 2.0**i;
  END LOOP;
  counter := 0.0;
  END PROCESS coder;
END ARCHITECTURE behaviour;
-----

```

DEFINITION DES
PARAMETRES :
-nombre de bits
DEFINITION DES
CONNECTIONS I/O :
-entrée différentielle
-code thermomètre

PROCESS&BOUCLE :

Sur chaque changement
du code thermomètre le
code binaire est recalculé.

Code 4 - 11 : Modèle en VHDL-AMS de l'encoder binaire

IV.9. Modèle du Convertisseur Numérique/Analogique

Ce modèle est sans complexité car basé sur une représentation idéale du convertisseur :

```

-----
ENTITY N_bits_thermo_DAC IS
  GENERIC (
    step      : REAL           := 0.80/15;
    N_bits    : INTEGER        := 4;
  )
  PORT (
    SIGNAL thermo_code : IN BIT_VECTOR(0 TO (2**N_bits-2));
    SIGNAL Noutp       : OUT REAL := -(2.0**(N_bits-1)-0.5)*step/2.0;
    SIGNAL Noutn       : OUT REAL := +(2.0**(N_bits-1)-0.5)*step/2.0;
  )
END ENTITY N_bits_thermo_DAC;
-----
ARCHITECTURE behaviour OF N_bits_thermo_DAC IS
BEGIN
  decoding : PROCESS (thermo_code)
  VARIABLE counter : REAL := 0.0;
  BEGIN
    FOR i IN 0 TO (2**N_bits)-2 LOOP
      IF thermo_code(i) = '1' THEN counter := counter + 1.0; END IF;
    END LOOP;
    Noutp <= +(-2.0**(N_bits-1) + 0.5 + counter)*step/2.0;
    Noutn <= -(-2.0**(N_bits-1) + 0.5 + counter)*step/2.0;
    counter := 0.0;
  END PROCESS decoding;
END ARCHITECTURE behaviour;
-----

```

*DEFINITION DES
PARAMETRES :*
-pas
-nombre de bits
*DEFINITION DES
CONNECTIONS I/O :*
-entrée code
thermométrique
- sortie différentielle

PROCESS &BOUCLE :
calcul de la valeur réelle
correspondant au code.

-passage de la sortie de
mono_voie à différentielle

Code 4 - 12 : Modèle en VHDL-AMS du CNA

IV.10. Modèle du résonateur

Par soucis de rapidité de simulation, la modélisation comportementale des AOT a été réduite à la saturation des sorties, le circuit différentiel est modélisé en monovoie, les fonctions de transfert en z.

```

-----
ENTITY complex_resonator IS
  GENERIC (
    VT      : REAL := 26.0e-3;
    IS_0    : REAL := 1.0e-15;
    C_FB    : REAL := 1.0765e-12;
    C_IN    : REAL := 8.0e-12;
    C_DAC   : REAL := 8.0e-12;
    C_CP    : REAL := 2.73e-12;
    C_M_1   : REAL := 4.0e-12;
    C_ER    : REAL := 0.9213e-12;
    C_DACR  : REAL := 0.9723e-12;
    C_EX    : REAL := 0.88846e-12;
    C_DACX  : REAL := 1.0412e-12;
    C_YRN   : REAL := 0.635e-12;
    C_YXN   : REAL := 1.46176e-12;
    C_M_2   : REAL := 2.0e-12;
    Vmax    : REAL := 1.0;
    Vmin    : REAL := -1.0;
  )
  PORT (
    TERMINAL VDD, VSS, MC : ELECTRICAL;
    TERMINAL IOPA, PIN    : ELECTRICAL_VECTOR ( 0 TO 3 );
    SIGNAL EN              : BIT;
    SIGNAL DAC             : IN REAL_VECTOR ( 0 TO 3 );
  )
END ENTITY complex_resonator;
-----

```

*DEFINITION DES
PARAMETRES :*
- tension de seuil
- courant de diode
- capacités
- tension de saturation
max et min des AOT

*DEFINITION DES
CONNECTIONS I/O :*
- alimentation en
tension et courant

```
SIGNAL CK_1, CK_2 : IN BIT_VECTOR ( 0 TO 7 );
SIGNAL OUT_2 : INOUT REAL_VECTOR ( 0 TO 3 ) := (1.65,1.65,1.65,1.65));
END complex_resonator;
```

ARCHITECTURE comportement **OF** complex_resonator **IS**

```
CONSTANT A_1 : REAL :=C_FB/C_M_1;
CONSTANT B_1 : REAL :=C_IN/C_M_1;
CONSTANT C_1 : REAL :=C_CP/C_M_1;
CONSTANT D_1 : REAL :=C_DAC/C_M_1;
CONSTANT A_2 : REAL :=C_YXN/C_M_2;
CONSTANT B_2 : REAL :=C_YRN/C_M_2;
CONSTANT C_2 : REAL :=C_ER/C_M_2;
CONSTANT D_2 : REAL :=C_DACR/C_M_2;
CONSTANT E_2 : REAL :=C_EX/C_M_2;
CONSTANT F_2 : REAL :=C_DACX/C_M_2;
SIGNAL ENA : REAL := 0.0;
QUANTITY Vmc ACROSS MC TO VSS;
QUANTITY Valim ACROSS lalim THROUGH VDD TO VSS;
QUANTITY Vin_I ACROSS PIN(0) TO PIN(1);
QUANTITY Vin_Q ACROSS PIN(2) TO PIN(3);
QUANTITY Vopa_1 ACROSS lopa_1 THROUGH VDD TO IOPA(0);
QUANTITY Vopa_2 ACROSS lopa_2 THROUGH VDD TO IOPA(1);
QUANTITY Vopa_3 ACROSS lopa_3 THROUGH VDD TO IOPA(2);
QUANTITY Vopa_4 ACROSS lopa_4 THROUGH VDD TO IOPA(3);
```

BEGIN

```
lopa_1 == Vopa_1/10.0e+3;
lopa_2 == Vopa_2/10.0 e+3;
lopa_3 == Vopa_3/10.0 e+3;
lopa_4 == Vopa_4/10.0 e+3;
lalim== (lopa_1+lopa_2+lopa_3+lopa_4)*20.0*ENA;
```

PROCESS (EN)

```
BEGIN IF (EN='1') THEN ENA <= 1.0; ELSE ENA <= 0.0; END IF;
```

END PROCESS;

filter: **PROCESS** (CK_2(0),CK_2(2))

```
VARIABLE Nin_I_z_1, Nin_Q_z_1 : REAL := 0.0;
VARIABLE Nout_1_I_z_1, Nout_1_Q_z_1 : REAL := 0.0;
VARIABLE Nout_2_I_z_1, Nout_2_Q_z_1 : REAL := 0.0;
VARIABLE Nout_2_I, Nout_2_Q : REAL := 0.0;
VARIABLE Nout_1_I, Nout_1_Q : REAL := 0.0;
```

BEGIN

IF ((CK_2(0)'event) **AND** (CK_2(0) = '0')) **THEN**

```
Nin_I_z_1 := Vin_I;
Nin_Q_z_1 := Vin_Q;
Nout_1_I_z_1 := Nout_1_I;
Nout_1_Q_z_1 := Nout_1_Q;
Nout_2_I_z_1 := OUT_2(0)-OUT_2(1);
Nout_2_Q_z_1 := OUT_2(2)-OUT_2(3);
```

END IF;

IF ((CK_2(2)'event) **AND** (CK_2(2) = '1')) **THEN**

XXXXXXXXXX Equations de la voie I&Q XXXXXXXXXXXXXXXXXXXX

```
IF Nout_1_I >= Vmax THEN Nout_1_I := Vmax; END IF;
```

```
IF Nout_1_I <= Vmin THEN Nout_1_I := Vmin; END IF;
```

```
IF Nout_1_Q >= Vmax THEN Nout_1_Q := Vmax; END IF;
```

```
IF Nout_1_Q <= Vmin THEN Nout_1_Q := Vmin; END IF;
```

```
IF Nout_2_I >= Vmax THEN Nout_2_I := Vmax; END IF;
```

```
IF Nout_2_I <= Vmin THEN Nout_2_I := Vmin; END IF;
```

```
IF Nout_2_Q >= Vmax THEN Nout_2_Q := Vmax; END IF;
```

```
IF Nout_2_Q <= Vmin THEN Nout_2_Q := Vmin; END IF;
```

- entrées différentielles
- enable
- horloges de commandes
- sortie différentielle I et Q

DEFINITION DES

CONSTANTES :

rapport de capacité
déterminant le gain des
sous montages

DEFINITION DU COUPLE

TENSION / COURANT
POUR LES NŒUDS
INTERNES ET
EXTERNNES

MISE EN EQUATIONS

-courant de consommation

- conversion de l'enable
de bit en real.

REVEIL DU PROCESS

SUR LES CLOCK CK2(0),
CK2(2)

-passage d'entrée
différentielle à mono-voie
- échantillonnages des
valeurs d'entrée et de
sortie

- équations voies I et Q
premier et second étage

- saturation des AOT

```

OUT_2(0) <= +Nout_2_I/2.0 + Vmc AFTER 10ns;
OUT_2(1) <= -Nout_2_I/2.0 + Vmc AFTER 10ns;
OUT_2(2) <= +Nout_2_Q/2.0 + Vmc AFTER 10ns;
OUT_2(3) <= -Nout_2_Q/2.0 + Vmc AFTER 10ns;
END IF;
END PROCESS filter;
END;

```

-passage de la sortie de
mono_voie à différentielle
avec retard

Code 4 - 13 : Modèle en VHDL-AMS du filtre résonateur complexe

IV.11. Modèle du modulateur Sigma Delta

```

ENTITY complex_sigma_delta_modulator IS
  GENERIC (
    noise_Vrms_sqrt_Hz : REAL := 15.0e-9;
    Tsmpt                : REAL := 1.0e-6;
    opt                  : REAL := 0.0;
    sigma_jitter         : REAL := 30.0e-1);
  PORT (
    TERMINAL VSS,VDD,MC : ELECTRICAL;
    TERMINAL IOPA       : ELECTRICAL_VECTOR ( 0 TO 3 );
    TERMINAL PIN        : ELECTRICAL_VECTOR ( 0 TO 3 );
    SIGNAL EN           : IN BIT := '1';
    SIGNAL clk_26M      : IN BIT := '1';
    SIGNAL code_out_I   : INOUT BIT_VECTOR(0 TO 3);
    SIGNAL code_out_Q   : INOUT BIT_VECTOR(0 TO 3));
END ENTITY complex_sigma_delta_modulator;

```

DEFINITION DES
CONNECTIONS I/O :
- alimentations
- courants de pol°
- entrées
- enable
- horloge entrant
- code voie I et Q

ARCHITECTURE behaviour_kernel OF complex_sigma_delta_modulator IS

```

CONSTANT d_up      : REAL := 1.97e-9;
CONSTANT d_down    : REAL := 1.82e-9;
CONSTANT d_0       : REAL := 1.0e-9;
CONSTANT d_1       : REAL := 0.5e-9;
CONSTANT d_2       : REAL := 1.0e-9;
CONSTANT N_bits    : INTEGER := 4;
CONSTANT step_Q    : REAL := (1.832-1.468)*2.0/15.0;
CONSTANT delay_Q   : REAL := 3.0e-9;
CONSTANT step_DAC  : REAL := 0.8/15.0;

SIGNAL thermol     : BIT_VECTOR (0 TO (2**N_bits)-2) := (OTHERS => '0');
SIGNAL thermoQ    : BIT_VECTOR (0 TO (2**N_bits)-2) := (OTHERS => '0');
SIGNAL DAC        : REAL_VECTOR (0 TO 3) := (0.0, 0.0,0.0,0.0);
SIGNAL Out_2      : REAL_VECTOR (0 TO 3) := (0.0, 0.0,0.0,0.0);
SIGNAL CK_2       : BIT_VECTOR (0 TO 7) := ('0','0','0','0','0','0','0','0');
SIGNAL Latch      : BIT := '0';

```

DEFINITION DES
PARAMETRES
INTERNES :
pas et nombre de bits du
CAN, CNA, générateur
d'horloges

BEGIN

```

clock_no : ENTITY no_clocks_gene (behaviour)
  GENERIC MAP ( delay_up => d_up, delay_down => d_down, delay_0 => d_0,
    Delay_1 => d_1, delay_2 => d_2, sigma => sigma_jitter)
  PORT MAP ( clock_in => clk_26M ,clock_out => CK_2, latch_ADC => latch);

```

GENERATEUR
D'HORLOGES

```

filter : ENTITY complex_resonator (behaviour)
  PORT MAP ( VSS=>VSS,VDD=>VDD,MC=>MC,IOPA=>IOPA,Pin=>Pin, EN=>
    EN,DAC=>DAC,CK_2=>CK_2,CK_1=>CK_2,OUT_2=>Out_2);

```

RESONATEUR
COMPLEXE

```

Quantifier_I : ENTITY N_bits_Quantifier (behaviour)
  GENERIC MAP ( step => step_Q, N_bits=>N_bits, delay => delay_Q)
  PORT MAP (latch=>latch,Ninp=>Out_2(0),Ninn=>Out_2(1),thermo_code=>thermol) ;

```

CAN/CNA/encodeur voie I
formant la boucle de
retour

```

DAConversion_I : ENTITY N_bits_thermo_DAC (behaviour)
  GENERIC MAP ( step => step_DAC, N_bits=>N_bits)
  PORT MAP ( Noutp=>DAC(0),Noutn=>DAC(1),thermo_code=>thermol);

```

```

Coder_I : ENTITY N_bits_Digital_Coder (behaviour)
GENERIC MAP ( N_bits => N_bits, option => '1')
PORT MAP ( thermo_code => thermoI, binary_code => code_out_I);

Quantifier_Q : ENTITY N_bits_Quantifier (behaviour)
GENERIC MAP ( step => step_Q, N_bits=>N_bits, delay => delay_Q)
PORT MAP (latch=>latch,Ninp=>Out_2(2),Ninn=>Out_2(3),thermo_code=>thermoQ) ;

DACConversion_Q : ENTITY N_bits_thermo_DAC (behaviour)
GENERIC MAP ( step => step_DAC, N_bits=>N_bits)
PORT MAP ( Noutp=>DAC(2),Noutn=>DAC(3),thermo_code=>thermoQ);

Coder_Q : ENTITY N_bits_Digital_Coder (behaviour)
GENERIC MAP ( N_bits => N_bits, option => '1')
PORT MAP ( thermo_code => thermoQ, binary_code => code_out_Q);

END ARCHITECTURE behaviour_kernel;

```

*CAN/CNA/encodeur voie
Q formant la boucle de
retour*

Code 4 - 14 : Modèle en VHDL-AMS du Sigma Delta Complexe

IV.12. Modèle de bruit de phase du générateur Rhode&Schwarz

```

FUNCTION Generator_noise (Fsmp, PSD_1, fc_1, PSD_2, PSD_3, fc_3 : REAL; int:
REAL_VECTOR(0 TO 18)) RETURN REAL_VECTOR IS

VARIABLE outt : REAL_VECTOR (0 TO 18) := int;

BEGIN

outt(14 TO 18) := white_noise (Fsmp, PSD_3, int(14 TO 18));
outt(13) := (MATH_PI*fc_3*(outt(14)+int(14))+int(13)*(Fsmp-MATH_PI*fc_3))/
(Fsmp+MATH_PI*fc_3);

outt(8 TO 12) := white_noise (Fsmp, PSD_2, int(8 TO 12));
outt(3 TO 7) := white_noise (Fsmp, PSD_1, int(3 TO 7));
outt(2) := (MATH_PI*fc_1*(outt(3)+int(3))+int(2)*(Fsmp-MATH_PI*fc_1))/
(Fsmp+MATH_PI*fc_1);
outt(1) := (MATH_PI*fc_1*(outt(2)+int(2))+int(1)*(Fsmp-MATH_PI*fc_1))/
(Fsmp+MATH_PI*fc_1);
outt(0) := outt(1)+outt(8)+outt(13);

RETURN (outt);
END FUNCTION;

END PACKAGE BODY noise_functions;

```

Code 4 - 15 : Modèle en VHDL-AMS du bruit de phase du générateur

Nous réalisons une simulation temporelle suivie d'une analyse fréquentielle : périodogramme sur $N = 1048576$ échantillons avec un fenêtrage Blackman7 ($F_{ech} = 20\text{MHz}$).

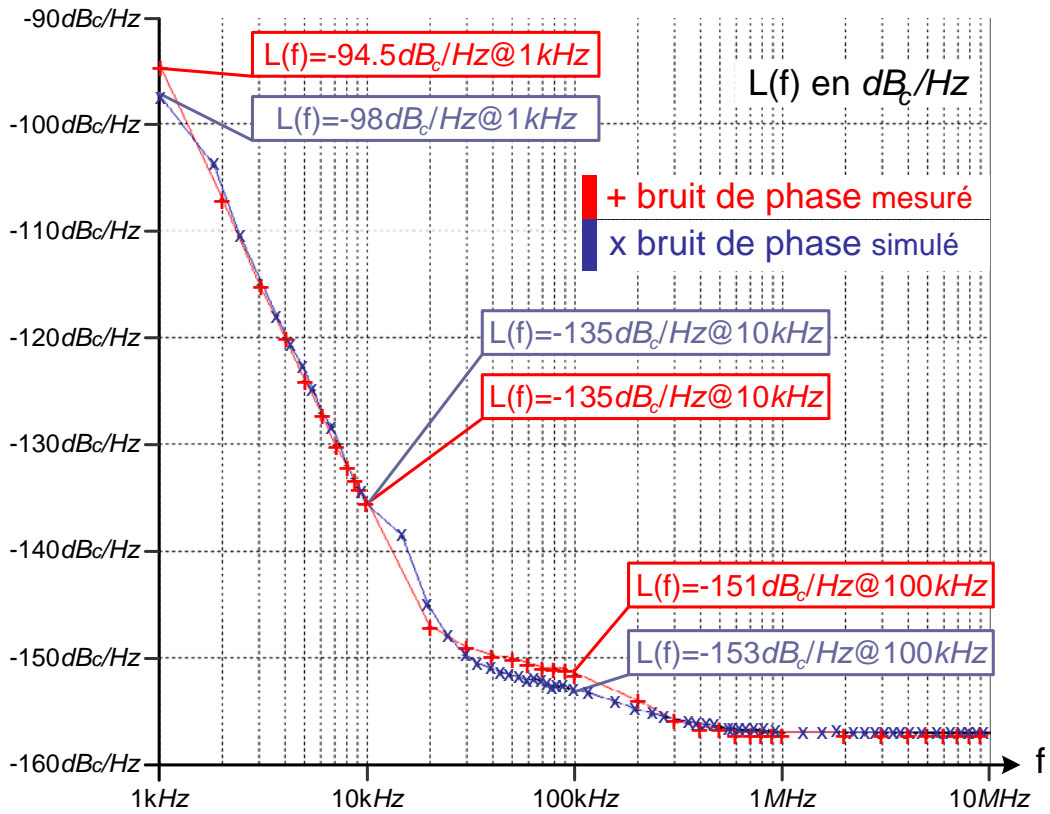


Figure 4 - 35 : Bruit de phase du générateur Rhode&Schwarz (modèle et mesure)

Résumé

Cette thèse concerne l'évaluation des performances en bruit des blocs analogiques et mixtes situés en début d'une chaîne de réception radiofréquence. Ce travail s'attache plus particulièrement à l'étude du modulateur Sigma Delta à temps discret utilisé pour la conversion analogique/numérique. Le but de cette étude est de fournir aux concepteurs de circuits radiofréquence des modèles comportementaux nécessaires à l'évaluation de l'impact des sources de bruit sur certaines architectures.

C'est dans cette optique que nous avons développé une bibliothèque en VHDL-AMS comportant des modèles de sources de bruit à temps discret et des modèles comportementaux de circuits analogiques et mixtes (Sigma Delta, bloc RF). Nous avons démontré la validité de la modélisation des sources de bruit par comparaison avec la théorie. Les autres modèles ont été étalonnés à la fois par comparaison avec des simulations au niveau transistor et par comparaison avec des mesures réalisées sur des composants industriels.

Le point critique de ces travaux étant l'estimation et la modélisation du jitter interne du modulateur Sigma Delta, nous avons développé conjointement une méthode spécifique de test, un circuit ainsi que sa carte d'évaluation. La valeur mesurée du jitter interne nous a permis de valider l'ensemble de la démarche, grâce à une simulation globale de la chaîne en situation réelle.

Abstract

This thesis is dedicated to the noise performance evaluation of the analog and mixed blocks that are located at the beginning of a radiofrequency receiver chain. The main subject of this work is to study a discrete time Sigma Delta modulator, used for analog to digital conversion. The goal is to provide the RF designers, behavioral models capable of evaluating the impact of noise sources on different architectures.

That is why we developed a library in VHDL-AMS for discrete time models of noise sources and for behavioral models of analog and mixed blocks (Sigma Delta modulator, RF blocks). We have proved the validity of the noise models by comparing our simulation and the theory. To calibrate the behavioral models, we have performed comparative simulations (transistor model vs. VHDL-AMS model) and measurements on industrial components.

The critical point was to estimate the Sigma Delta internal jitter characteristics. This is the reason why we have developed a test circuit, a specific evaluation board associated to an original method. This test method has permitted the internal jitter estimation. We have validated our approach with a global simulation of the physical test bench.

Mots-clés

Circuit Radio Fréquence analogiques et mixtes
Modulateur Sigma Delta à temps discret
Modélisation comportementale VHDL-AMS
Bruit dans les architectures à temps discret

Key words

Radio-Frequency analog and mixed circuit
Discrete time Sigma Delta modulator
VHDL-AMS behavioural models
Noise in discrete time architecture