Traitement du signal échantillonné non uniformément : algorithme et architecture

F. Aeschlimann

▶ **To cite this version:**

F. Aeschlimann. Traitement du signal échantillonné non uniformément : algorithme et architecture. Micro et nanotechnologies/Microélectronique. Institut National Polytechnique de Grenoble - INPG, 2006. Français. NNT : . tel-00011758

HAL Id: tel-00011758

https://theses.hal.science/tel-00011758v1

Submitted on 6 Mar 2006

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

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

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
|__/__/__/__/__/__/__/__/__/__|

THESE

pour obtenir le grade de

DOCTEUR DE l'INPG

Spécialité : Signal Image Parole Télécom

préparée au laboratoire **TIMA** dans le cadre de
l'Ecole Doctorale d'« Electronique, Electrotechnique, Automatique, Télécommunications, Signal »

présentée et à soutenue publiquement

par
Fabien AESCHLIMANN

Le 6 février 2006

Titre :

TRAITEMENT DU SIGNAL ÉCHANTILLONNÉ NON UNIFORMÉMENT : ALGORITHME ET ARCHITECTURE

Directeur de Thèse : Marc RENAUDIN
Codirecteur : Laurent FESQUET

JURY

M. Gang FENG	, Président
M. Jacques OKSMAN	, Rapporteur
M. Olivier SENTIEYS	, Rapporteur
M. Marc RENAUDIN	, Directeur de thèse
M. Laurent FESQUET	, Co-directeur de thèse
M. Dominique SEBILLE	, Examinateur

A Juliette, sans qui je ne serais jamais venu à Grenoble,

A mon père, sans qui je ne serais jamais venu à l'électronique,

A ma mère, sans qui je ne serais simplement jamais venu au monde!

Résumé

Ce travail de thèse s'intègre dans le cadre du développement de nouvelles approches de conception afin de réduire significativement la consommation électrique des Systèmes sur Puce (SoC) ou des Objets Communicants utilisés pour traiter numériquement des signaux. Le but est alors d'obtenir des systèmes entièrement contrôlés par les événements contenus dans les signaux. Dans ce contexte, une nouvelle catégorie de chaîne de traitement est définie, associant une implémentation matérielle asynchrone (sans horloge globale) et un échantillonnage non uniforme dans le temps dit « par traversée de niveaux ». Un convertisseur Analogique/Numérique dédié à cette tâche ayant déjà été réalisé, ce travail se focalise sur le traitement des données composées de couples amplitude-temps dont cette thèse montre que toute opération doit obligatoirement prendre en compte l'information temporelle. Des filtres numériques à réponse impulsionnelle finie (RIF) et infinie (RII) sont alors définis dans le cadre de signaux échantillonnés non uniformément. Des architectures sont proposées puis comparées à celles utilisées classiquement montrant que la complexité combinatoire était accrue. Un critère sur le choix de la technologie à privilégier, spécifiant la charge de calcul totale sur une durée finie, montre alors qu'en diminuant le nombre de points traités, l'approche asynchrone peut compenser le surcoût de complexité. Ainsi le traitement de signaux faiblement actifs par une chaîne asynchrone, combinant échantillonnage non uniforme et conception asynchrone, permet de réduire son activité moyenne et donc la consommation du circuit intégré, rendant cette technologie très attractive pour le domaine des SoC.

Mots-clés : circuits asynchrones, échantillonnage non uniforme par traversée de niveaux, schéma d'échantillonnage, filtre RIF, filtre RII.

Abstract

This PhD thesis deals with the development of new design approaches in order to reduce significantly the power consumption of Systems on Chips (SoC) and Communicating Objects used in digital signal processing. The goal is to obtain systems only driven by the events contained in the useful signal. In this context, a new kind of signal processing chain is proposed, combining an asynchronous design (no global clock) and a non-uniform sampling scheme called level-crossing sampling. As an analog-to-digital converter dedicated to this task has already been studied, this work is focussed on the sampled signal processing based on amplitude-time couples. A preliminary study shows that any operation has to use the temporal information. Then, Finite and Infinite Impulse Response Filters (FIR and IIR) are defined in the case of non-uniform sampled signals. Architectures are also proposed and compared to those commonly used showing that the computational complexity is increased. A criterion about the choice of the technology to favour, specifying the global computational load over a finite time, has proved that the asynchronous approach can compensate the complexity cost by decreasing the number of processed points. Thus the processing of low-active signals by an asynchronous chain, combining asynchronous design and non-uniform lead to a reduction of its average activity and so of the power consumption of the integrated circuits making this technology very attractive for Soc area.

Key words: asynchronous circuits, non-uniform level-crossing sampling, FIR filter, IIR filter, architectures

Table des matières

Résumé 5
Abstract 6
Liste des tableaux 13
Liste des figures 14
INTRODUCTION 25
CHAPITRE I Etat de l'art 31
I.1 Théorie de l'échantillonnage 32
I.1.1 Fonction et schéma d'échantillonnage 32
I.1.2 Transformée de Fourier discrète généralisée 32
I.1.3 Echantillonnage uniforme 33
I.1.4 Echantillonnage non uniforme 34
I.1.4.1 Echantillonnage non uniforme à *P* fréquences 36
I.1.4.2 Echantillonnage uniforme avec jitter 38
I.1.4.3 Echantillonnage uniforme avec perte d'échantillons 39
I.1.4.4 Echantillonnage par traversée de niveaux 40
I.2 Principes des circuits asynchrones 41
I.2.1 Comparaison des circuits synchrones et asynchrones 42
I.2.1.1 Rapidité de calcul 42
I.2.1.2 Consommation d'énergie 42
I.2.1.3 Modularité 43
I.2.1.4 Emissions électromagnétiques 43
I.2.2 Protocole de communication 44
I.2.2.1 Protocole à deux phases 44
I.2.2.2 Protocole à quatre phases 45
I.2.2.3 La porte de Müller 46
I.2.3 Catégories de circuits asynchrones 46
I.2.3.1 Circuits de Huffman 47
I.2.3.2 Circuits Micropipelines 47
I.2.3.3 Circuits Indépendants de la Vitesse 49
I.2.3.4 Circuits Insensibles aux Délais 49
I.2.3.5 Circuits Quasi Insensibles aux Délais 50

I.2.4 Convertisseurs analogique numérique asynchrones 50
I.3 Conclusion : combinaison d'un échantillonnage non uniforme et d'une conception asynchrone 51
I.4 Présentation du Convertisseur A/N Asynchrone : le CANA 52
I.4.1 Principe de fonctionnement du convertisseur 52
I.4.2 Quantification du temps et Rapport Signal sur Bruit 53
I.4.3 Implémentation asynchrone du CANA 56
I.4.4 Non linéarité de la conversion asynchrone 58
CHAPITRE II Etude de l'échantillonnage par traversée de niveaux des signaux périodiques 59
II.1 Etude de cas : échantillonnage d'un signal sinusoïdal 60
II.1.1 Etude des intervalles de temps 60
II.1.1.1 Intervalles de temps des variations croissantes 61
II.1.1.2 Intervalles de temps des variations décroissantes 61
II.1.1.3 Intervalles de temps des crêtes du signal 62
II.1.2 Facteur de symétrie 63
II.1.2.1 Evolution du facteur de symétrie autour d'un niveau 66
II.1.2.2 Evolution du facteur de symétrie autour d'un inter-niveau 68
II.1.2.3 Evolution du facteur de symétrie dans son ensemble 70
II.1.2.4 Conclusion : impact de la symétrie sur l'échantillonnage 73
II.1.3 Exemples de représentations d'un signal sinusoïdal 74
II.1.4 Schéma d'échantillonnage et spectre d'un signal sinusoïdal 75
II.2 Généralisation : échantillonnage d'un signal périodique quelconque 78
II.2.1 Etude des intervalles de temps 79
II.2.2 Symétrie 79
II.2.3 Exemples de représentations d'un signal périodique 80
II.2.4 Schéma d'échantillonnage et spectre d'un signal périodique 81
CHAPITRE III Etude de l'échantillonnage par traversée de niveaux des signaux non périodiques 85
III.1 Echantillonnage d'un signal impulsionnel 86
III.1.1 Choix du signal 86
III.1.2 Etude des intervalles de temps 87
III.1.3 Schéma d'échantillonnage et spectre d'un signal impulsionnel 88
III.1.3.1 Schéma d'échantillonnage 88
III.1.3.2 Spectre d'un signal impulsionnel échantillonné 89

III.1.3.3 Schéma d'échantillonnage d'un signal impulsionnel avec saturation du timer 93
III.1.4 Extension aux signaux impulsionnels périodiques 95
III.2 Etude d'un signal non stationnaire 98
III.2.1.1 Schéma d'échantillonnage 99
III.2.1.2 Exemple de représentation temps-fréquence d'un signal non périodique 100
III.3 Conclusion : l'échantillonnage et le repliement de spectre 102
CHAPITRE IV Reconstruction des signaux échantillonnés non uniformément 103
IV.1 Reconstruction d'un signal à temps continu 104
IV.1.1 Reconstruction idéale 104
IV.1.2 Reconstruction pratique 108
IV.1.2.1 Reconstruction par bloqueur d'ordre 0 108
IV.1.2.2 Reconstruction par interpolation d'ordre 1 109
IV.2 Etude de la distorsion du signal reconstruit 111
IV.2.1 Analyse spectrale : comparaison avec la GDFT 111
IV.2.2 Influence du quantum 113
IV.2.3 Equivalence avec un convertisseur classique 115
IV.3 Conclusion 116
CHAPITRE V Filtrage numérique à réponse impulsionnelle finie 117
V.1 Introduction 118
V.2 Produit de convolution asynchrone 120
V.2.1 Produit de convolution analogique 120
V.2.2 Produit de convolution asynchrone 121
V.2.2.1 Définition 121
V.2.2.2 Condition de fin de calcul 122
V.2.2.3 Initialisation 122
V.2.2.4 Choix des interpolations 123
V.2.2.5 Interpolation des points manquants 123
V.2.2.6 Calcul d'une intégrale numérique 124
V.2.2.6.1 Intégrale simplifiée 124
V.2.2.6.2 Intégrale complète 125
V.2.2.7 Algorithme itératif 125
V.2.2.8 Complexité combinatoire 127
V.2.2.9 Erreur de calcul 129

V.2.2.10 Produit de convolution asynchrone d'ordre 0 129
V.2.2.10.1 Algorithme 130
V.2.2.10.2 Complexité combinatoire 130
V.2.2.10.3 Erreur de calcul 131
V.2.2.11 Produit de convolution asynchrone d'ordre 0.5 132
V.2.2.11.1 Algorithme simplifié 132
V.2.2.11.2 Algorithme complet 134
V.2.2.11.3 Complexités combinatoires 136
V.2.2.11.4 Erreurs de calcul 138
V.2.2.12 Produit de convolution asynchrone d'ordre 1 140
V.2.2.12.1 Algorithme simplifié 140
V.2.2.12.2 Algorithme complet 142
V.2.2.12.3 Complexités combinatoires 144
V.2.2.12.4 Erreurs de calcul 145
V.2.2.13 Conclusion : comparaison des méthodes 146
V.3 Filtre numérique RIF asynchrone 148
V.3.1 Réponse impulsionnelle 148
V.3.2 Complexité combinatoire 148
V.3.3 Réponse en fréquence 149
V.3.4 Exemples de filtrage 151
V.4 Architecture matérielle 155
V.4.1 Choix de la structure 155
V.4.2 Structure itérative dans le cas uniforme 155
V.4.3 Structure itérative dans le cas non uniforme 156
V.4.3.1 Registre à décalage 157
V.4.3.2 Mise à jour des données 159
V.4.3.3 Adressage des registres 160
V.4.3.4 Latence de l'architecture 161
V.5 Conclusion 162
CHAPITRE VI Filtrage numérique à réponse impulsionnelle infinie 165
VI.1 Introduction 166
VI.1.1 Filtrage RII de signaux échantillonnés régulièrement 166
VI.1.2 Mise en équations 166

VI.1.3 Limitation dans le cas non uniforme 166
VI.2 Représentation d'un filtre analogique dans l'espace d'état 167
VI.2.1 Vecteur d'état 167
VI.2.2 Matrices d'état 168
VI.3 Représentation d'un filtre numérique dans l'espace d'état 170
VI.3.1 Méthodes standard 171
VI.3.1.1 Méthode d'Euler progressive 171
VI.3.1.2 Méthode d'Euler rétrograde 171
VI.3.1.3 Méthode bilinéaire 171
VI.3.1.4 Résolution de l'équation différentielle 172
VI.3.2 Conditions de stabilité et flot de conception 173
VI.3.2.1 Conditions de stabilité 173
VI.3.2.1.1 Définition 173
VI.3.2.1.2 Stabilité de la représentation d'état à temps continu 173
VI.3.2.1.3 Stabilité de la représentation d'état à temps discret 174
VI.3.2.2 Flot de conception 175
VI.3.3 Schémas numériques généralisés 176
VI.3.3.1 Schémas explicites et implicites 176
VI.3.3.2 Schéma explicite Runge-Kutta 4 177
VI.3.3.3 Schéma semi-implicite de Runge-Kutta 3 178
VI.4 Réponse en fréquence du filtre numérique 179
VI.5 Exemples de filtrage 182
VI.6 Architectures matérielles 187
VI.6.1 Analyse des complexités combinatoires des schémas numériques 187
VI.6.1.1 Echantillonnage uniforme 187
VI.6.1.2 Echantillonnage non uniforme 187
VI.6.1.2.1 Structures simples du 1er et 2ème ordre 188
VI.6.1.2.2 Schéma de Runge-Kutta d'ordre 4 188
VI.6.1.3 Comparaison 190
VI.6.2 Architectures 191
VI.6.2.1 Architecture d'un filtre utilisant la méthode bilinéaire 191
VI.6.2.2 Architecture d'un filtre utilisant le schéma RK4 192
VI.7 Conclusion 196

CHAPITRE VII Evaluation d'une chaîne de traitement numérique du signal asynchrone 197
VII.1 Choix d'une application 198
VII.2 Mise en œuvre d'une chaîne de traitement asynchrone 199
VII.2.1 Conversion Analogique/Numérique Asynchrone 199
VII.2.2 Conception du filtre numérique 199
VII.2.2.1 Implémentation d'un filtre RIF 200
VII.2.2.2 Implémentation d'un filtre RII 200
VII.2.3 Résultats 201
VII.3 Critère de choix de la technologie 204
VII.4 Conclusion 210
CONCLUSION 211

Liste des tableaux

Tableau 1 : Récapitulatif des origines d'un signal échantillonné non uniformément.....35
Tableau 2 : Caractéristiques comparées des convertisseurs 56
Tableau 3 : Récapitulatif des erreurs maximales en fonction de la résolution du convertisseur 114
Tableau 4 : Comparaison des caractéristiques des différents produits de convolution asynchrone147
Tableau 5 : Complexité combinatoire d'un filtre RII échantillonné régulièrement.....187
Tableau 6 : Complexités combinatoires des schémas décomposables en une structure simple 188
Tableau 7 : Complexité combinatoire du schéma de Runge-Kutta d'ordre 4.....190
Tableau 8 : Augmentation de la complexité combinatoire par rapport au cas régulier en fonction de l'ordre *N* (pour *N* grand) 190
Tableau 9 : Cycles de calcul de l'architecture intégrant le schéma RK4.....195
Tableau 10 : Cahier des charges du CANA pour une application de parole 199
Tableau 11 : Rapport minimum entre le nombre de points synchrones et le nombre de points asynchrones privilégiant la technologie asynchrone quel que soit l'ordre du filtre 206
Tableau 12 : Gains extrapolés en fonction de l'activité du signal de parole présenté dans notre exemple pour les différents filtres.....209
Tableau 13 : Synthèse des caractéristiques des filtres étudiés dans ce travail de thèse.....215

Liste des figures

Figure 1 : Echantillonnage non uniforme à *P* fréquence (*P = 3*) 36
Figure 2 : Echantillonnage uniforme avec un jitter 38
Figure 3 : Echantillonnage uniforme avec perte d'échantillons 39
Figure 4 : Echantillonnage non uniforme par traversée de niveaux 40
Figure 5 : Schéma fonctionnel d'un circuit asynchrone 42
Figure 6 : Protocole à deux phases 45
Figure 7 : Protocole à quatre phases 45
Figure 8 : Symbole et table de vérité d'une porte de Müller à 2 entrées 46
Figure 9 : Catégories de circuits asynchrones 47
Figure 10 : Contrôle local d'un circuit Micropipeline 48
Figure 11 : Circuit Micropipeline avec traitement 49
Figure 12 : Equivalence entre les circuits SI et QDI 50
Figure 13 : Boucle de conversion asynchrone 52
Figure 14 : Quantification du temps induite par le timer lors de la traversée d'un niveau 54
Figure 15 : Schéma fonctionnel asynchrone du CANA 56
Figure 16 : Layout et photographie du convertisseur CANA 58
Figure 17 : Cycle hystérésis des intervalles de temps autour d'une crête 62
Figure 18 : Evolution temporelle des intervalles de temps en fonction de la valeur moyenne signal 64
Figure 19 : Représentation *tête-bêche* des deux crêtes d'un signal sinusoïdal en fonction de sa valeur moyenne 65
Figure 20 : Evolutions schématiques des intervalles de temps des crêtes et du facteur de symétrie en fonction de la valeur moyenne 71
Figure 21 : Exemple d'évolution simulée des intervalles de temps en fonction de la valeur moyenne 72
Figure 22 : Exemple d'évolution simulée du facteur de symétrie en fonction de la valeur moyenne 73
Figure 23 : Amplitude d'un signal sinusoïdal échantillonné par traversée de niveaux. 74
Figure 24 : Intervalles de temps d'un signal sinusoïdal échantillonné par traversée de niveaux. 74
Figure 25 : Schémas d'échantillonnage de signaux sinusoïdaux 76
Figure 26 : Plan large du schéma d'échantillonnage d'un signal sinusoïdal 76

Figure 27 : Spectre de signaux sinusoïdaux échantillonnés par traversée de niveaux 77
Figure 28 : Produit de convolution entre le spectre d'un signal sinusoïdal à temps continu et son schéma d'échantillonnage par traversée de niveaux 78
Figure 29 : Exemple de signal périodique symétrique échantillonné symétriquement 80
Figure 30 : Exemple d'un signal périodique symétrique échantillonné non symétriquement 81
Figure 31 : Exemple d'un signal périodique non symétrique. L'échantillonnage est également non symétrique. 81
Figure 32 : Spectre d'un signal périodique échantillonné par traversée de niveaux 83
Figure 33 : Echantillonnage par traversée de niveaux d'un signal impulsionnel 88
Figure 34 : Schéma d'échantillonnage normalisé d'un signal impulsionnel 90
Figure 35 : Spectre d'un signal impulsionnel échantillonné par traversée de niveaux 91
Figure 36 : Rapport de la largeur du lobe principal du schéma d'échantillonnage sur la bande passante du signal en fonction de la résolution du convertisseur 92
Figure 37 : Fonction d'échantillonnage d'un signal impulsionnel avec saturation du timer 93
Figure 38 : Représentation temps-fréquence théorique du schéma d'échantillonnage d'un signal impulsionnel avec saturation du timer 94
Figure 39 : Représentation temps-fréquence théorique du schéma d'échantillonnage d'un signal impulsionnel variable de période T_0 avec saturation du timer 96
Figure 40 : Echantillonnage par traversée de niveaux de deux impulsions décalées. Des périodes inactives induisent la saturant du timer. 97
Figure 41 : Représentations temps fréquence à bande étroite (à gauche) et large bande (à droite) en lignes de niveaux du schéma d'échantillonnage de deux impulsions décalées 98
Figure 42 : Transition entre le peigne de Dirac en fréquence et le schéma d'échantillonnage d'une impulsion 98
Figure 43 : Fréquence instantanée d'une modulation linéaire de fréquence 99
Figure 44 : Allure théorique du schéma d'échantillonnage et du spectre d'une modulation linéaire de fréquence échantillonnée 100
Figure 45 : Exemple de modulation linéaire de fréquence 101
Figure 46 : Représentation temps-fréquence du schéma d'échantillonnage et du spectre du signal échantillonné dans le cadre d'une modulation linéaire de fréquence 101
Figure 47 : Interprétation du théorème de Beutler dans le cadre de l'échantillonnage stationnaire (figure de gauche) et non stationnaire (figure de droite) d'un signal non stationnaire 108
Figure 48 : Reconstruction par bloqueur d'ordre 1 dans le cas uniforme (a) et non uniforme (b) 111

Figure 49 : Comparaison entre les spectres obtenus par GDFT (à gauche) et par TF du signal reconstruit à l'ordre 0 (à droite) 112
Figure 50 : Erreur relative sur l'amplitude du fondamental en fonction du nombre de points échantillonnés par cycle d'un signal sinusoïdal 113
Figure 51 : Taux de rejection de l'harmonique 2 en fonction du nombre de points échantillonnés par cycle d'un signal sinusoïdal 114
Figure 52 : Effet du lissage d'un signal reconstruit par bloqueur d'ordre 0 dans le cas régulier (à gauche) et dans le cas non uniforme par traversée de niveaux (à droite) 116
Figure 53 : Architecture de filtrage RIF à temps continu 120
Figure 54 : Exemple du produit de convolution entre une fonction porte et l'échelon d'Heavyside 121
Figure 55 : Synchronisation des signaux à l'initialisation d'un calcul 122
Figure 56 : Echantillonnage des points manquants par interpolation 124
Figure 57 : Calcul simplifié de l'intégrale par multiplication des échantillons et utilisation de trapèzes élémentaires 125
Figure 58 : Calcul complet de l'intégrale par multiplication des fonctions interpolées 125
Figure 59 : Organigramme générique de l'algorithme itératif 127
Figure 60 : Organigramme du produit de convolution asynchrone d'ordre 0 131
Figure 61 : Etude du pire cas de l'ordre 0 131
Figure 62 : Produit de convolution asynchrone d'ordre 0.5s 133
Figure 63 : Organigramme du produit de convolution asynchrone d'ordre 0.5s 134
Figure 64 : Produit de convolution asynchrone d'ordre 0.5c 135
Figure 65 : Organigramme du produit de convolution asynchrone d'ordre 0.5c 137
Figure 66 : Etude du pire cas de l'ordre 0.5s 138
Figure 67 : Etude du pire cas de l'ordre 0.5c 139
Figure 68 : Produit de convolution asynchrone d'ordre 1s 141
Figure 69 : Organigramme du produit de convolution asynchrone d'ordre 1s 142
Figure 70 : Produit de convolution asynchrone d'ordre 1c 143
Figure 71 : Organigramme du produit de convolution asynchrone d'ordre 1c 144
Figure 72 : Etude du pire cas de l'ordre 1 145
Figure 73 : Représentation de l'erreur relative maximale normalisée en fonction de la complexité combinatoire pour chaque méthode 147

Figure 74 : Filtrage RIF passe-bas à 33 coefficients de fréquence de coupure $f_c = 3Hz$ et échantillonnés à $F_e = 20Hz$ selon les cinq interpolations (plan large et zoom)..... 152
Figure 75 : Filtrage RIF passe-haut à 33 coefficients de fréquence de coupure $f_c = 3Hz$ et échantillonnés à $F_e = 20Hz$ selon les cinq interpolations (plan large et zoom)..... 153
Figure 76 : Evolution du nombre de surfaces totales dans le produit de convolution asynchrone d'ordre 0.5c en fonction de la fréquence d'échantillonnage de la réponse impulsionnelle.... 153
Figure 77 : Réponses impulsionnelles des filtres passe-bas et passe-haut..... 154
Figure 78 : Filtrages à phase linéaire d'un signal sinusoïdal..... 154
Figure 79 : Structure itérative d'un filtre RIF dans le cas uniforme..... 156
Figure 80 : Structure itérative d'un filtre RIF dans le cas non uniforme..... 157
Figure 81 : Architecture du bloc *RD ax* : registre à décalage des amplitudes du signal d'entrée... 158
Figure 82 : Architecture du bloc *MAJ* : mise à jour des intervalles de temps 160
Figure 83 : Architecture du bloc *ADD* : mise à jour des adresses des registres 161
Figure 84 : Représentation dans l'espace d'état d'un filtre analogique..... 170
Figure 85 : Transformation du demi-plan gauche avec la méthode d'Euler rétrograde 175
Figure 86 : Transformation du demi-plan gauche pour la méthode d'Euler progressive 175
Figure 87 : Transformation du demi-plan gauche pour la méthode bilinéaire 175
Figure 88 : Flot de conception d'un filtre numérique stable..... 176
Figure 89 : Transformation inverse du cercle unité pour la méthode Runge-Kutta d'ordre 4..... 178
Figure 90 : Transformation inverse du cercle unité pour la méthode de Runge-Kutta d'ordre 3 à 2 étages..... 179
Figure 91 : Projection de l'axe imaginaire du plan complexe analogique par les transformations des méthodes bilinéaire (a), d'Euler rétrograde et progressive (b), Runge-Kutta d'ordre 4 (c) et RK23 (d) 181
Figure 92 : Exemple de réponses en fréquence déformées par la méthode bilinéaire pour deux valeurs de fréquence d'échantillonnage (les périodisations ne sont affichées pour plus de clarté) 182
Figure 93 : Représentation dans le plan complexe analogique des pôles des filtres de Butterworth d'ordre 6 (à gauche) et de Chebyshev d'ordre 4 (à droite)..... 183
Figure 94 : Filtre de Butterworth d'ordre 6 (plan large et zoom) 184
Figure 95 : Filtre de Chebyshev d'ordre 4 (plan large et zoom)..... 184
Figure 96 : Filtre de Butterworth d'ordre 10 déstabilisant la méthode RK4 (plan large et zoom). 185
Figure 97 : Filtre de Chebyshev d'ordre 10 déstabilisant la méthode RK4 (plan large et zoom) .. 186

Figure 98 : Représentation dans le plan complexe analogique des pôles des filtres de Butterworth d'ordre 10 (à gauche) et de Chebyshev pour les ordres 2 à 10 (à droite) 186
Figure 99 : Structure cascadée d'un filtre RII utilisant la méthode bilinéaire 191
Figure 100 : Structures du 1er ordre et du 2ème ordre utilisant la méthode bilinéaire 192
Figure 101 : Structure cascadée d'un filtre RII utilisant le schéma de Runge-Kutta d'ordre 4 193
Figure 102 : Structures *A* et *B* utilisées par le schéma de Runge-Kutta d'ordre 4 193
Figure 103 : Contrôleur asynchrone de la structure *A* utilisé par le schéma de Runge-Kutta d'ordre 4 194
Figure 104 : Contrôleur asynchrone de la structure *B* utilisé par le schéma de Runge-Kutta d'ordre 4 195
Figure 105 : Gabarit du filtre passe bas 200
Figure 106 : Réponse en fréquence du filtre RIF 201
Figure 107 : Réponse en fréquence (à gauche) et pôles (à droite) du filtre RII 201
Figure 108 : Filtrage numérique d'un signal de parole échantillonné non uniformément par traversée de niveaux 202
Figure 109 : Transformées de Fourier des signaux reconstruits à l'ordre 0 202
Figure 110 : Evolutions temporelles du nombre d'itérations et d'échantillons d'entrée utilisés dans le produit de convolution asynchrone. 204
Figure 111 : Gain d'une chaîne de traitement numérique asynchrone en fonction du rapport entre le nombre d'échantillons synchrones et asynchrones. Pour le filtrage RII, les courbes de gains sont paramétrées par l'ordre du filtre. Il y a donc une série de courbes pour chaque méthode; les courbes tendent toutes vers une droite limite. 207
Figure 112 : Rapport minimum entre le nombre de points synchrones et le nombre de points asynchrones en fonction de l'ordre des filtres 207
Figure 113 : Gains extrapolés en fonction de l'activité du signal de parole présenté dans notre exemple pour les différents filtres 209

Remerciements

Durant quatre années, j'ai eu l'honneur de travailler au laboratoire TIMA (*Techniques de l'Informatique et de la Microélectronique pour l'Architecture des ordinateurs*) de *l'Institut National Polytechnique de Grenoble* (INPG). Je remercie son directeur, Bernard Courtois, de m'avoir accueilli au sein de ce laboratoire.

Je remercie Marc Renaudin, mon directeur de thèse, Professeur à *l'Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Grenoble* (ENSERG) et responsable du groupe de recherche *Concurent Integrated Systems*, de m'avoir accepté au sein de son équipe et de m'avoir fait partager sa passion pour les approches nouvelles en électronique que sont notamment les circuits asynchrones. Ces années ont été très enrichissantes tant sur le plan technique que sur le plan humain.

Je remercie également Laurent Fesquet, mon co-directeur de thèse et Maître de Conférence à *l'Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Grenoble* et Brigitte Bidégaray, Chargée de Recherche au *Laboratoire de Modélisation et de Calcul* (LMC), pour leur disponibilité et leurs éclairages techniques et mathématiques durant ces années.

Je remercie Jacques Oksman, Professeur à *l'Ecole Supérieure d'Electricité* (SUPELEC) et Olivier Sentieys, Professeur à *l'Ecole Nationale Supérieure des Sciences Appliquées et de Technologie* (ENSSAT), d'avoir accepté d'être les rapporteurs de mes travaux de thèse. Je remercie également Dominique Sebille, directeur R&D de la branche *Commutation et Systèmes de Détection de Valeo* de participer à mon jury de thèse et de l'intérêt qu'il porte à ces travaux et des perspectives industrielles qu'il pourra amener. Je remercie enfin Gang Feng, Professeur à *l'Ecole Nationale Supérieure d'Electronique et de Radioélectricité de Grenoble* d'avoir accepté de présider le jury de thèse.

Je tiens par ailleurs à remercier Guillaume Gibert et Joäo Fragoso, mes deux compagnons d'arme avec qui j'ai partagé nombres d'aventures passionnantes… Guillaume tout d'abord où chaque midi (restau U puis café chez Jacky), il écouta patiemment et avec attention mes élucubrations les plus diverses, et ce depuis quatre ans. Quatre ans, déjà. Ces longues discussions n'étaient pas que de pures pertes de temps; elles servaient aussi à faire le point sur nos sujets respectifs et grâce à cela, plusieurs verrous ont pu être levés. Joäo ensuite qui depuis le même laps de temps, traîne à mes cotés que ce soit dans la cave ou en N135, cette véritable usine à

champions… J'ai rarement rencontré quelqu'un d'aussi sage (sage comme un bonze, certainement pas sage comme une image !). Ses conseils ont toujours été très pertinents et il m'a fait progresser dans de nombreux domaines notamment en informatique où les problèmes sont généralement dus à *l'interface chaise-clavier* et en bureautique bien que lui-même ne sache pas comment imprimer une section particulière d'un document word.

Comment ne pas remercier également Emmanuel Allier, précurseur du traitement numérique du signal échantillonné non uniformément ? Ses excellents travaux étant à la base des miens, il serait injuste de ne pas lui rendre un brillant hommage… Mais au fait Manu… Il vient d'où le 8.2kHz ?

Comment ne pas remercier non plus les autres membres du groupe CIS, le groupe qui ferait passer Les Bronzés pour des rigolos… Si Les Bronzés vont à la mer, au ski et maintenant en Italie tous les 25 ans, le groupe CIS, c'est : des pauses cafés, des pots d'anniversaire, des pots de thèse alcoolisés, des crémaillères, des restos, des sorties au cinéma, des soirées crêpes, des soirées Halloween, des soirées boite-de-nuit, des soirées et encore des soirées… Mais le groupe CIS, c'est aussi des sorties : des sorties karting, des sorties laser-game, des randonnées, du vélo, de la montagne et du ski !! Bref, avec le groupe CIS quand tu ne trouves rien pour ta recherche, il y en a toujours d'autres qui te trouve un truc pour le week-end…

C'est donc parti pour remercier l'ensemble du groupe en commençant par Gilles, le dernier chef et co-responsable du Master Pro CSINA qui fait des avions en papier avec les copies de ses élèves (faut dire qu'elles ne méritent pas meilleur sort !); puis viennent les anciens du groupe : JB, qui aurait pu être une marque de Whisky, Ahn-Vu qui pour l'avoir vérifié, le pensait vraiment, Amine qui est toujours resté d'une gentillesse remarquable, Kamel que je n'ai jamais réussi à avoir avec mon gobelet, Dhanistha qui croit à tout même au fait de risquer la mort en buvant une canette (j'en profite également pour remercier les conjoints et donc son *chouchou* de Fred et de ses bons plans à la Julien Courbet), Antoine qui nous a bien fait rire, Jérôme-j'ai-de-la-chance qui a réussi à se casser la main un jeudi 12 alors que je lui avait prédit une armoire le 13 (et Pascale dont le père a des fourches attirées par la tête de Jérôme), Salim qui en bon musulman rigole aux mêmes blagues que moi, Arnaud qui m'a appris beaucoup de choses sur les techniques de drague et d'approche et enfin Pascal et Edith qui m'ont supporté (dans le deux sens) comme ils le pouvaient à Async'04…; viennent ensuite ceux qui sont encore là : Yannick qui quand il n'est pas en réunion à l'ENSERG jusqu'à 11h, boit de la bière et/ou joue de la guitare, Fraidy qui peut être assimilé à un

spécimen humain du marsupilami, Beber que j'ai réussi à décapiter au couteau, Vivi qui nous met tous les jours une ambiance de feu avec ses claquettes et ses t-shirts, Estelle qui a des peluches très ouvertes sur les choses de l'amour (et son Christophe qui voyait ma femme tous les matins… dans le bus en allant au boulot…), Grégory que j'ai eu comme élève et qui n'a plus beaucoup de points sur sa copie à force de me dénigrer, Yann qui m'a montré que la photographie, ce n'est pas simplement appuyer sur un bouton, Aurélien qui s'il ne sait pas quoi faire après la thèse pourra toujours postuler soit chez VideoGag soit chez JackAss, David qui parait-il me mettrait la pâté avec son aïkido, Livier qui fait des gâteaux d'enfer dont un cheese cake à la pêche, je vous en parle pas, Cédric qui ne trouve rien d'autre de mieux que de monter, à la suite, 5 des plus durs cols des alpes pour s'amuser et enfin Nicolas, le remplaçant du remplaçant d'Arnaud qui s'est offert un an de vacances dans l'hémisphère sud, de quoi de me donner des idées… Mais le groupe CIS, c'est aussi une communauté élargie donc des membres apparentés : Sophie et son chat qu'elle laisse martyriser par certains, Greg qui prend toujours deux maillots de rugby différents pour être sûr de ne pas jouer avec moi à l'entraînement (et Céline qui ne sait pas faire la bouffe pour dix sans la faire pour cent), Yannick qui est passé depuis qu'il fait de la boxe, de 1m10 de haut à 1m10 d'épaisseur (et Mimine qui peut vous montrer ce qu'elle a mangé juste en buvant quelques verres de rhum…), Pierre sorte de Jérôme en plus chanceux enfin Caro et Damien qui se prennent pour des écrivisses à chaque fois qu'ils sont près de l'eau… Finalement, le groupe CIS c'est aussi des nouveaux chaque année : Jérémie, Eslam et Saeed à qui je souhaite bonne chance pour leurs thèses surtout à ce dernier qui aura la dure tâche de prendre la suite de mon travail et le mettre en valeur.

Je voudrais dans un dernier élan remercier l'ensemble des personnes que j'ai côtoyées pendant ma thèse. Les pom pom girls du TIMA : Isabelle, Corinne, Joëlle, Anne Laure, Chantal, Lucie, Ahmed, les Sébastien… Olivier Rosseto qui a accepté gentiment de me prêter son cours de traitement du signal; Alexandre Chagoya, Bernard Bosc et Robin Rolland pour leur extrême gentillesse; les élèves de mes trois promotions du Master Pro CSINA, de l'ENSERG et du département Télécom pour m'avoir supporté comme prof (faut dire que je les supportais bien comme élèves !).

Merci à tous, chaleureusement...

Avant-propos

Bien que la spécialité de cette thèse s'inscrive dans le domaine du traitement du signal, l'ensemble du travail a été réalisé au sein d'un laboratoire de microélectronique dans la perspective d'une réalisation matérielle avancée d'une chaîne de traitement numérique du signal en technologie asynchrone. Afin de rendre ce manuscrit le plus agréable possible à la lecture, en particulier pour les non spécialistes i.e. les microélectroniciens, le formalisme mathématique a été réduit au minimum et certains termes habituellement utilisés en traitement du signal ont été supprimés pour des raisons de clarté. En espérant que chacun puisse trouver, malgré tout, les informations nécessaires à son propre travail, nous vous souhaitons à tous une bonne lecture.

INTRODUCTION

La tendance actuelle en microélectronique est d'intégrer des systèmes complexes sur une seule puce (« SoC » pour « System on Chip ») : capteurs, convertisseurs Analogique/Numérique (CAN), traitements numériques du signal, transmissions de données. Or malgré une complexité de plus de plus en plus croissante, les contraintes imposées lors de leur conception sont de plus en plus fortes. En effet, ces systèmes doivent être peu coûteux, portables, faiblement rayonnants et surtout peu consommateurs d'énergie parce qu'ils sont souvent alimentés par des batteries. La forte consommation d'énergie constitue la plupart du temps le principal point bloquant de leur conception.

Le but de cette thèse est donc de repenser complètement la chaîne classique de traitement numérique du signal afin de réduire son activité et du même coup la consommation électrique du circuit dans lequel elle est intégrée. Du de vue point matériel, dans la perspective de la réalisation de ces systèmes intégrés très complexe, notre groupe de recherche travaille sur la conception de « circuits asynchrones ». Alors que le fonctionnement des circuits synchrones est orchestré par un signal global unique (horloge), les circuits asynchrones sont contrôlés par une multitude de signaux qui gèrent les échanges d'information entre blocs fonctionnels. Le fonctionnement est ainsi de type flot de données ce qui offre à cette catégorie de circuits intégrés des propriétés intéressantes : modularité, exécution en temps minimum, faible bruit et faible consommation puisque seul le matériel impliqué dans le calcul consomme de l'énergie ; la mise en veille du circuit est inhérente au principe asynchrone ce qui facilite la gestion d'énergie. Par ailleurs du point de vue du signal, bien que la plupart des systèmes traitent des signaux dont les propriétés statistiques sont intéressantes, aucune architecture de traitement du signal n'en tient vraiment compte. En réalité, ces signaux (comme la température, la pression, les électrocardiogrammes, la parole…) sont généralement constants mais peuvent varier fortement pendant une courte période. Ainsi, l'échantillonnage classique à pas constant, que l'on appellera échantillonnage régulier ou uniforme, et le système de conversion sont hautement contraints par le théorème de Shannon qui doit assurer que la fréquence d'échantillonnage soit au moins égale au double de la fréquence maximale du

signal. L'information des échantillons prélevés dans les périodes stables est alors redondante entraînant une suractivité du circuit et donc un surcoût élevé en énergie.

Le travail mené dans le cadre de cette thèse est donc focalisé sur le traitement du signal en vue de la réalisation de fonctions en logique asynchrone. L'objectif est de se libérer de l'échantillonnage régulier au profit d'un échantillonnage non uniforme contrôlé par le signal lui-même dit « par traversée de niveaux ». Nous remettons en cause la chaîne classique de traitement, « conditionnement du signal », « échantillonnage » et « traitement numérique synchrone », car elle ne prend pas en compte la nature de l'information contenue dans le signal. C'est pourquoi en combinant la conception en logique asynchrone et l'échantillonnage par traversée de niveaux, une nouvelle classe de chaîne de traitement numérique du signal est définie sur la base d'un convertisseur Analogique/Numérique Asynchrone (CANA). La chaîne traite ainsi des échantillons prélevés en fonction des variations du signal d'entrée, associant une information sur l'amplitude et une information temporelle caractérisant la non uniformité, donc le signal. Cet échantillonnage est parfaitement adapté au traitement numérique asynchrone car, comme lui, il est contrôlé par des événements dont les instants d'occurrence sont a priori inconnus. On sort d'ici du cadre restrictif de la conception d'une algorithmique prévue pour le traitement des signaux échantillonnés régulièrement destinés à s'exécuter sur des circuits synchrones. La maîtrise de la conception de machines asynchrones nous amène donc à considérer une algorithmique basée sur une autre représentation des signaux numériques. L'étude théorique de l'échantillonnage par traversée de niveaux pour différentes classes de signaux montre que l'information temporelle est nécessaire lors d'un traitement afin de pondérer l'amplitude des échantillons. Des outils mathématiques ont ainsi été développés et mis en œuvre afin de réaliser des opérations de base spécifiques à l'échantillonnage non uniforme : transformée de Fourier, filtrage à réponse impulsionnelle finie et infinie. Les traitements sont formalisés, analysés en terme d'erreur et de complexité combinatoire et comparés au cas régulier ; des architectures matérielles en sont alors déduites en vue de leur intégration. Les simulations fonctionnelles montrent qu'en dépit d'une complexité accrue, le traitement asynchrone de signaux peu actifs permet une forte réduction de la consommation électrique du circuit par rapport au cas synchrone du fait de l'absence de redondance dans l'information traitée. Il apparaît clairement que les bénéfices de cette approche seront exploitables pour la conception de circuits intégrés complexes ouvrant donc des perspectives nouvelles pour la réalisation de « capteurs intelligents » et de systèmes de communication.

L'organisation de ce manuscrit de thèse est la suivante :

Le chapitre I présente les travaux antérieurs qui sont à la base des contributions réalisés au cours de cette thèse en synthétisant les études qui nous ont conduit à définir une nouvelle forme de chaîne de traitement du signal. Il se découpe en trois sections principales dans lesquelles nous présentons successivement les principes et concepts que nous avons utilisés afin de réduire l'activité d'une chaîne de traitement et sa consommation électrique. Tout d'abord, nous présentons l'intérêt d'un échantillonnage non uniforme dans le temps. Nous résumons les différentes formes classiques d'échantillonnages irréguliers et leurs propriétés. Parmi elles, nous mettons en valeur l'échantillonnage par traversée de niveaux qui lié au signal d'entrée permet d'adapter l'activité de la chaîne à l'activité du signal. Dans un second temps, nous présentons les principes d'une conception matérielle asynchrone c'est-à-dire sans l'utilisation d'une horloge globale, et les avantages qui en découlent. Nous décrivons les différents types de circuits possibles et leur fonctionnement. Utilisée dans le cadre d'une conversion analogique numérique, une conception asynchrone peut permettre de réduire le bruit, les métastabilités et la consommation électrique du circuit. Enfin, nous présentons le premier élément de la chaîne de traitement du signal : le convertisseur Analogique/Numérique Asynchrone ou CANA : le principe de cette nouvelle classe de convertisseur est détaillé : il s'agit de structures basées sur la combinaison simultanée d'une conception matérielle asynchrone et d'un échantillonnage non uniforme par traversée de niveaux. Cette alliance permet d'obtenir une convergence entre l'implémentation matérielle d'un convertisseur et son mode d'échantillonnage.

Le chapitre II présente l'échantillonnage par traversée de niveaux. Celui-ci étant un processus non linéaire dépendant du signal d'entrée, l'étude formelle de la conversion de signaux usuels doit être réalisée. Le choix des signaux usuels à analyser doit prendre en considération leurs variations car c'est en fonction de leurs évolutions que l'échantillonnage est commandé. Ainsi, nous proposons dans ce chapitre d'étudier la conversion de signaux périodiques car leurs dérivées ont la propriété d'être également périodiques. Dans un premier temps, une étude de cas est réalisée à partir d'un signal sinusoïdal car il permet facilement d'analyser les instants d'échantillonnage et les intervalles de temps en fonction de ses paramètres – fréquence fondamentale, amplitude crête et amplitude moyenne – pour un convertisseur donné. Puis dans un second temps, une généralisation à l'ensemble des signaux périodiques est proposée. Dans chaque cas le schéma d'échantillonnage, i.e. la modification effectué sur le spectre du signal analogique par l'échantillonnage, est caractérisé montrant que le spectre du signal à temps discret est replié.

Le chapitre III étend l'étude de l'échantillonnage par traversée de niveaux à des signaux non périodiques. En effet, nous voulons analyser les mécanismes d'adaptation de l'échantillonnage lorsque les propriétés statistiques de leurs dérivées varient au cours du temps. Dans un premier temps, nous considérons un signal impulsionnel gaussien et un signal composé d'une succession d'impulsions. Nous montrons que les propriétés du schéma d'échantillonnage évoluent au cours du temps : il bascule en effet alternativement entre un peigne de Dirac régulier pendant les zones inactives et un peigne de Dirac non uniforme qui dépend de l'impulsion. Dans un second temps, nous considérons un signal non stationnaire. Pour se ramener à un cas connu, nous utilisons une modulation linéaire de fréquence. Nous généralisons le cas précédent en montrant que le schéma d'échantillonnage s'adapte automatiquement au contenu spectral de la zone en cours de conversion. Dans tous les cas, le schéma d'échantillonnage conduit à nouveau à un repliement du spectre du signal échantillonné ce qui nous permet de conclure que le traitement d'un signal échantillonné non uniformément par traversée de niveaux doit utiliser obligatoirement l'information portée par les intervalles pour pondérer la valeur des amplitudes.

Le chapitre IV est dédié la reconstruction d'un signal à temps continu à partir d'un signal à temps discret échantillonné non uniformément. Nous proposons un récapitulatif historique relatant les principaux résultats classés chronologiquement car la majorité des publications qui sont dédiées aux signaux échantillonnés non uniformément, traitent de la reconstruction idéale. Puis, nous étudions la reconstruction pratique, c'est-à-dire par le biais d'un bloqueur d'ordre 0 et d'ordre 1. Nous montrons, à partir du spectre du signal reconstruit, que la perte d'information, mise en évidence dans les chapitres précédents, par un repliement de spectre systématique du signal échantillonné, peut être compensée par l'utilisation de l'information contenue dans les intervalles de temps. Par ailleurs, nous montrons que le spectre du signal reconstruit par bloqueur d'ordre 0 et d'ordre 1 ne dépend que de l'amplitude et de l'intervalle de temps des échantillons. Il est donc possible d'utiliser une approximation numérique du spectre du signal reconstruit à la place d'une transformée de Fourier Discrète pour réaliser une analyse spectrale du signal échantillonné.

Le chapitre V est dédié au traitement des données échantillonnées non uniformément par traversée de niveaux. Parmi le vaste choix de traitements possibles, nous décidons d'étudier le filtrage à réponse impulsionnelle finie. En effet, le filtrage est certainement l'application la plus élémentaire que l'on peut effectuer sur un signal – aussi bien avec des signaux à temps continu qu'à temps discret. Nous choisissons le filtrage à réponse impulsionnelle finie, car parmi les filtres numériques, ce sont les plus simples à réaliser : d'une part la sortie ne dépend que des échantillons

d'entrée – il s'agit du produit de convolution entre l'entrée et la réponse impulsionnelle – et d'autre part, il sont toujours stables. La première partie est donc consacrée à la définition d'un produit de convolution à temps discret dédié aux signaux échantillonnés non uniformément. Celui-ci consiste à calculer à partir de signaux échantillonnés non uniformément, un produit de convolution analogique à un instant donné en interpolant les signaux à temps discret, en les multipliant puis en calculant leur intégrale. Afin de ne pas expliciter l'interpolation, nous introduisons un algorithme itératif qui décompose le résultat en une somme d'aires élémentaires obtenues directement à partir des échantillons. Nous étudions pour des interpolations simples les algorithmes, leurs complexités combinatoires ainsi que l'erreur maximale obtenue dans le pire cas. Dans la seconde partie, nous définissons un filtre à réponse impulsionnelle finie basé sur le produit de convolution utilisant une interpolation d'ordre 0. Nous étudions ses caractéristiques (réponse impulsionnelle, réponse en fréquence) et en déduisons une architecture matérielle détaillée.

Nous montrons, lors d'un filtrage, qu'une relation entre des échantillons d'entrée et des échantillons de sortie peut être établie en définissant un produit de convolution entre le signal d'entrée et la réponse impulsionnelle. Cette méthode est dédiée à l'utilisation de réponses impulsionnelles finies qui limitent la durée du calcul. En revanche, pour la catégorie des filtres à réponses impulsionnelles infinies, cette technique n'est, par définition, plus applicable. La solution que nous présentons dans le chapitre VI est basée sur la représentation d'un filtre analogique dans l'espace d'état et sa discrétisation par un schéma numérique. Les principes de cette technique puis de nouveaux schémas numériques sont introduits. La stabilité d'un filtre numérique est alors étudiée en fonction du schéma. Dans un second temps, nous nous intéressons à l'implémentation matérielle d'un filtre numérique à réponse impulsionnelle infinie. Nous comparons les complexités combinatoires de chaque méthode afin d'éliminer les schémas nécessitant les coûts les plus importants. Nous en déduisons alors deux architectures asynchrones différentes pour deux schémas distincts : la méthode bilinéaire, inconditionnellement stable et la méthode Runge-Kutta d'ordre 4, conditionnellement stable.

Le chapitre VII est dédié à l'évaluation d'une chaîne de traitement numérique du signal asynchrone. Dans un premier temps, un exemple est considéré pour illustrer la mise en œuvre d'une chaîne. Il concerne le filtrage d'un signal de parole dont l'activité est réduite à 25% du temps total (75% de silence). Puis dans un second temps, nous définissons un critère général de comparaison entre une chaîne synchrone et asynchrone. Il permet de déterminer la technologie à privilégier en fonction du traitement considéré. Dans chaque cas, la charge de calcul totale, utilisée

pour traiter une série de points échantillonnés pendant une durée finie, est prise en compte. Comme un traitements asynchrone est plus complexe que le même traitement réalisé dans le cas synchrone, il doit, pour être avantageux, s'appliquer sur un nombre de points réduits afin que, sur une durée donnée, la charge de calcul totale soit plus faible que dans le cas asynchrone – le nombre de points limités compensant une complexité accrue. Des résultats numériques sont ensuite présentés dans le cadre des filtres mis en ouvre dans les chapitres précédents. Ils montrent qu'à partir d'un rapport minimal entre le nombre de points de la chaîne synchrone, échantillonnés régulièrement, et le nombre de points de la chaîne asynchrone, échantillonnés non uniformément, un traitement asynchrone est préconisé quel que soit l'ordre du filtre considéré; la charge de calcul totale sur une durée limitée qui, dans tous les cas est plus faible, induit une réduction de l'activité de la chaîne et donc de sa consommation électrique.

CHAPITRE I
Etat de l'art

Ce chapitre rassemble les travaux antérieurs qui sont à la base des contributions réalisées au cours de cette thèse en synthétisant les études qui nous ont conduits à définir une nouvelle forme de chaîne de traitement du signal. Il se découpe en trois sections principales dans lesquelles nous présentons successivement les principes et concepts que nous avons utilisés afin de réduire l'activité d'une chaîne de traitement et sa consommation électrique.

Tout d'abord, nous présentons l'intérêt d'un échantillonnage non uniforme dans le temps. Nous résumons les différentes formes classiques d'échantillonnages irréguliers et leurs propriétés. Parmi elles, nous mettons en valeur l'échantillonnage par traversée de niveaux qui, lié au signal d'entrée, permet d'adapter l'activité de la chaîne à l'activité du signal.

Dans un second temps, nous présentons les principes d'une conception matérielle asynchrone c'est-à-dire sans l'utilisation d'une horloge globale, et les avantages qui en découlent. Nous décrivons les différents types de circuits possibles et leur fonctionnement. Utilisée dans le cadre d'une conversion analogique numérique, une conception asynchrone peut permettre de réduire le bruit, les métastabilités et la consommation électrique du circuit.

Enfin nous présentons le premier élément de la chaîne de traitement du signal : le convertisseur Analogique/Numérique Asynchrone ou CANA. Le principe de cette nouvelle classe de convertisseur est détaillé : il s'agit de structures basées sur la combinaison simultanée d'une conception matérielle asynchrone et d'un échantillonnage non uniforme par traversée de niveaux. Cette alliance permet d'obtenir une convergence entre l'implémentation matérielle d'un convertisseur et son mode d'échantillonnage.

I.1 Théorie de l'échantillonnage

I.1.1 Fonction et schéma d'échantillonnage

En traitement du signal, l'échantillonnage permet de représenter un signal à temps continu par un signal à temps discret. Il est l'opération de base du traitement numérique sans laquelle aucune application n'est envisageable. Dans le domaine temporel, il consiste à multiplier la fonction à temps continu *x(t)* par une somme d'impulsions de Dirac translatées à des instants différents, appelée *fonction d'échantillonnage FE(t)*.

$$FE(t) = \sum_{n \in \mathbf{Z}} \delta(t - t_n) \qquad \text{Eq. (1)}$$

Le signal échantillonné peut alors simplement s'écrire :

$$x_E(t) = \sum_{n \in \mathbf{Z}} x(t_n)\delta(t - t_n) \qquad \text{Eq. (2)}$$

Dans le domaine fréquentiel, l'échantillonnage correspond au produit de convolution entre les transformées de Fourier du signal à temps continu et de la fonction d'échantillonnage. Cette dernière appelée *schéma d'échantillonnage* s'obtient à partir de l'équation (1) par linéarité :

$$SE(f) = \sum_{n \in \mathbf{Z}} e^{-j2\pi f t_n} \qquad \text{Eq. (3)}$$

D'après l'équation précédente, la forme du schéma d'échantillonnage est caractérisée par l'ensemble des instants d'échantillonnage. Ainsi selon la forme du peigne de Dirac, donc selon la technique d'échantillonnage utilisée, le schéma d'échantillonnage et le spectre du signal échantillonné $X_E(f)$ seront différents :

$$X_E(f) = \sum_{n \in \mathbf{Z}} x(t_n) e^{-j2\pi f t_n} \qquad \text{Eq. (4)}$$

I.1.2 Transformée de Fourier discrète généralisée

La transformée de Fourier discrète généralisée (en anglais General Discrete Fourier Transform – GDFT) permet de calculer numériquement le spectre d'un signal échantillonné à partir d'une suite de *N* points consécutif [Bagshaw *et al.* 1991] :

$$\widetilde{X}_E(m) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-j2\pi m \Delta f t_n} \qquad \text{Eq. (5)}$$

où l'axe des fréquences est discrétisé avec un pas Δf quelconque, défini par l'utilisateur. Sachant que le signal échantillonné est tronqué sur N points par un fenêtrage rectangulaire, la GDFT n'est qu'une approximation du spectre théorique (comme la Transformée de Fourier Discrète). Celui-ci est en effet convolué par la transformée de Fourier de la fenêtre, c'est à dire dans le cas présent, par un sinus cardinal. L'échantillonnage fréquentiel peut alors introduire des fuites d'énergie en fonction du choix de N et de Δf. Pour un signal périodique par exemple, il faut s'assurer que la largeur de la fenêtre soit un multiple entier de la période fondamentale et que la fréquence fondamentale soit également un multiple entier du pas fréquentiel.

La GDFT parfois appelée aussi *Non-Uniform Discrete Fourier Transform* (NUDFT) a fait l'objet depuis une décennie d'études pour une implémentation rapide. Nous pouvons citer notamment les travaux de [Dutt *et al.* 1993], de [Duijndam *et al.* 1999] ou encore plus récemment ceux de [Greengard *et al.* 2004].

I.1.3 Echantillonnage uniforme

Le traitement numérique du signal actuel repose sur un échantillonnage uniforme des signaux à temps continu. Il produit ainsi un signal à temps discret dont les échantillons sont régulièrement espacés d'une durée T_e, la période d'échantillonnage. Les équations (2), (3) et (4) deviennent respectivement :

$$x_E(t) = \sum_{n\in\mathbf{Z}} x(nT_e)\delta(t - nT_e) \qquad \text{Eq. (6)}$$

$$SE(f) = \frac{1}{T_e}\sum_{n\in\mathbf{Z}} \delta(f - nF_e) \qquad \text{Eq. (7)}$$

$$X_E(f) = \frac{1}{T_e}\sum_{n\in\mathbf{Z}} X(f - nF_e) \qquad \text{Eq. (8)}$$

Le schéma d'échantillonnage est un peigne de Dirac uniforme en fréquence. Le spectre du signal échantillonnage est donc périodisé à tous les multiples entiers de la fréquence d'échantillonnage. Ainsi il est possible de retrouver facilement le théorème de Shannon (également

appelé *WKS theorem* en hommage aux deux autres contributeurs Whittaker et Kotelnikov) qui découle de la périodisation : un signal à temps continu à bande limitée, de fréquence maximale F_{max}, peut être reconstruit à partir d'observations régulières si l'occurrence de ces observations respecte la condition :

$$F_e \geq 2F_{max} \qquad \text{Eq. (9)}$$

En effet, en respectant cette condition, la périodisation n'introduit pas de repliement entre les motifs; le spectre du signal analogique, c'est-à-dire le signal analogique lui-même peut être obtenu en filtrant le signal échantillonné par un filtre passe bas idéal dont la réponse impulsionnelle est un sinus cardinal :

$$x(t) = \sum_{n \in \mathbf{Z}} x(nT_e)\,\mathrm{sinc}\left(\pi F_e \left(t - nT_e\right)\right) \qquad \text{Eq. (10)}$$

Ce résultat fondamental du traitement du signal, qui montre qu'aucune information n'est perdue lors de l'échantillonnage si celui-ci est suffisamment dense, avait déjà été formulé en premier lieu par Cauchy puis par Nyquist dans les années 1920 ; c'est pourquoi, on parle souvent de la fréquence de Nyquist pour spécifier le double de la fréquence maximale : $F_{NYQ} = 2F_{max}$.

Par ailleurs, l'expression de la GDFT se simplifie dans le cas d'un échantillonnage uniforme pour retrouver la formulation classique de la Transformée de Fourier Discrète en posant $t_n = nT_e$ et $\Delta f = F_e/N$.

I.1.4 Echantillonnage non uniforme

Dans le cas d'un échantillonnage non uniforme, les instants d'échantillonnage sont quelconques. La fonction d'échantillonnage reste donc sous la forme générale décrite par l'équation (1). Il existe de nombreuses études sur l'échantillonnage non uniforme car en pratique le cas uniforme n'est jamais respecté. L'échantillonnage qu'il soit *naturel* ou implémenté dans un système de conversion (on parlera de signaux *volontairement* échantillonnés) induit toujours une incertitude dans la connaissance des instants d'échantillonnage. Lorsque les données proviennent d'observations ou de mesures réalisées sans capteurs, c'est-à-dire lorsque le signal est naturellement échantillonné, les conditions dans lesquelles sont prélevés les échantillons ne peuvent assurer la régularité des occurrences. On peut citer notamment en astronomie des irrégularités des observations liées aux conditions météorologiques, à l'orbite de la terre ou bien à

des pannes techniques; De la même manière en géophysique, lors du carottage d'une couche de glace par exemple, la variation de hauteur de neige évolue d'une année sur l'autre rendant non uniforme l'échantillonnage de la carotte et de la base temporelle associée. Parallèlement lorsque les données sont volontairement échantillonnées par un système de conversion, la présence d'un signal d'horloge déclenchant l'échantillonnage ne peut garantir une précision infinie des instants d'échantillonnage. Bien que généralement négligée, la fréquence de l'horloge subit toujours de petites variations appelées *gigue* (*jitter* en anglais) dont l'importance varie en fonction du système oscillant. Une autre source naturelle d'irrégularité apparaît sur des données volontairement échantillonnées : la perte d'échantillon lors de transmission de données sur le canal voire sur le support (rayures sur un disque laser par exemple).

Parfois, l'échantillonnage peut être volontairement non uniforme, c'est-à-dire imposé par l'utilisateur. Toutefois, ce cas n'intervient pas lorsque signal est naturellement échantillonné car si les conditions le permettent, les observations se feront toujours régulièrement. En revanche, lorsque le signal est échantillonné par un dispositif, il se peut qu'il soit volontairement non uniforme pour compresser le signal par exemple. Enfin, l'échantillonnage peut également être volontairement non uniforme afin de modéliser l'influence des irrégularités d'un échantillonnage naturel et d'en déduire des techniques de traitement appropriées comme une reconstruction.

Le tableau suivant résume les différents types d'échantillonnages non uniformes possibles et leurs origines.

		Signal échantillonné	
		Par nature	Volontairement
Signal non uniforme	Par nature	Mesures physiques sans capteurs, observations…	Jitter de l'horloge, perte d'échantillons lors d'une transmission…
	Volontairement		Convertisseurs non uniformes, modélisations …

Tableau 1 : Récapitulatif des origines d'un signal échantillonné non uniformément

Le traitement d'un signal échantillonné irrégulièrement diffère selon l'origine de la non uniformité : dans la majorité des cas, l'irrégularité étant naturelle, la reconstruction d'un signal à

temps continu est principalement recherchée d'où l'importance de cette thématique dans la littérature. La modélisation de signaux échantillonnés non uniformément s'ajoute à cette catégorie car elle permet de valider les algorithmes et d'en évaluer les performances. En revanche lorsque l'irrégularité est forcée par le système, la reconstruction du signal à temps continu n'a pas grand intérêt. Seul le traitement des données est envisagé en terme d'analyse spectrale ou de filtrage. Nous nous plaçons dans ce cas là.

I.1.4.1 Echantillonnage non uniforme à *P* fréquences

L'échantillonnage non uniforme à P fréquences consiste à prélever des échantillons avec des intervalles de temps variables, appartenant à un ensemble de *P* périodes d'échantillonnage possibles : $\{F_{ei} = 1/T_{ei}\}$ avec $i \in [1, ..., P]$. A chaque fréquence est associée une probabilité d'apparition $p(i)$.

Chaque occurrence se produit après une durée τ_n de la précédente. Les instants d'échantillonnage sont donc une combinaison linéaire des différentes périodes d'échantillonnage possibles :

$$t_n = t_{n-1} + \tau_n = \sum_{i=1}^{P} \alpha_i T_{ei} \text{ avec } \alpha_i \in \mathbf{N}^+ \qquad \text{Eq. (11)}$$

Les coefficients α_i représentent le nombre de fois qu'a été utilisée chaque fréquence. La Figure 1 est un exemple dans lequel le signal est échantillonné par trois fréquences différentes.

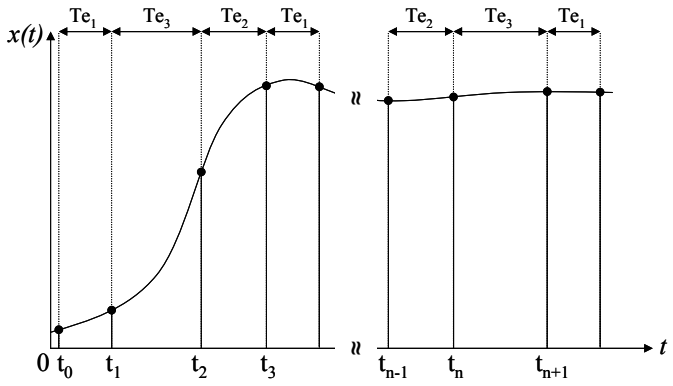


Figure 1 : Echantillonnage non uniforme à *P* fréquence (*P = 3*)

Bagshaw a montré que le schéma d'échantillonnage peut être périodique sous certaines conditions [Bagshaw *et al.* 1991]. A partir de l'équation (3), un schéma est périodique de période F_P s'écrivant :

$$SE(f+F_P) = \sum_{n\in\mathbf{Z}} e^{-j2\pi(f+F_P)t_n} = \sum_{n\in\mathbf{Z}} e^{-j2\pi ft_n} e^{-j2\pi F_P t_n} \qquad \text{Eq. (12)}$$

si $e^{-j2\pi F_P t_n} = 1$ pour tout $n \in \boldsymbol{Z}$, c'est-à-dire si $F_P t_n \in \boldsymbol{N}$. D'après l'équation (11), cette condition devient : $F_P \sum_{i=1}^{P} \alpha_i T_{ei} \in \boldsymbol{N}$. Or, comme les coefficients α_i sont entiers positifs, une condition nécessaire et suffisante peut s'écrire :

$$F_P T_{ei} \in \boldsymbol{N} \text{ pour tout } i \in [1, \ldots, P] \qquad \text{Eq. (13)}$$

Pour déterminer la valeur de la période du schéma d'échantillonnage en fonction des périodes d'échantillonnage, Bagshaw suppose que les périodes sont des nombres rationnels $\{T_{ei} = a_i/b_i\}$ avec $i \in [1, \ldots, P]$ et $\{a_i\}, \{b_i\} \in \boldsymbol{N}^+$. Cette simplification permet d'exprimer directement F_P en fonction de a_i et b_i :

$$F_P = \frac{ppcm\{b_i\}}{pgcd\{a_i\}} \qquad \text{Eq. (14)}$$

où $ppcm\{\}$ représente le plus petit commun multiple et $pgcd\{\}$, le plus grand commun diviseur. En posant $P = 1$, nous retrouvons directement la périodicité F_e du schéma d'échantillonnage uniforme. L'avantage d'utiliser un échantillonnage non uniforme à P fréquences est d'augmenter la fréquence de périodisation du schéma d'échantillonnage et donc de pouvoir analyser, numériquement avec la GDFT, le spectre d'un signal à temps continu sans pour autant respecter le théorème de Shannon. En effet plus le nombre de fréquences d'échantillonnage augmente, plus la périodisation du spectre est repoussée : il est alors possible d'analyser un signal dont la fréquence maximale est plus grande que chacune des fréquences d'échantillonnage prises séparément. Il s'agit de la propriété anti-repliement de l'échantillonnage non uniforme (en anglais *Alias-Free Sampling*).

I.1.4.2 Echantillonnage uniforme avec jitter

Comme nous l'avons présenté précédemment, un système de conversion basé sur un échantillonnage uniforme introduit naturellement une erreur sur les instants d'échantillonnage comme le montre la Figure 2. Causée par les fluctuations de la fréquence de l'horloge commandant l'échantillonnage, celle-ci est généralement négligée dans la plupart des situations pour conserver les techniques habituelles de traitement numérique du signal. Cependant, une fluctuation de la fréquence d'échantillonnage peut présenter un avantage : si la fréquence d'échantillonnage peut prendre n'importe quelle valeur dans une bande donnée, le schéma d'échantillonnage n'est pas périodique; il n'y a pas de repliement de spectre. Ce résultat démontré en 1960 par Shapiro et Silverman [Shapiro *et al.* 1960] est à la base d'une série d'applications appelée en anglais *Alias-Free* : l'analyse spectrale de signaux de plusieurs GHz [Tarczynski *et al.* 2004], [Tarczynski *et al.* 2005] ou bien l'amélioration des techniques de transmission radiofréquence [Wojtiuk 2000] en sont quelques exemples.

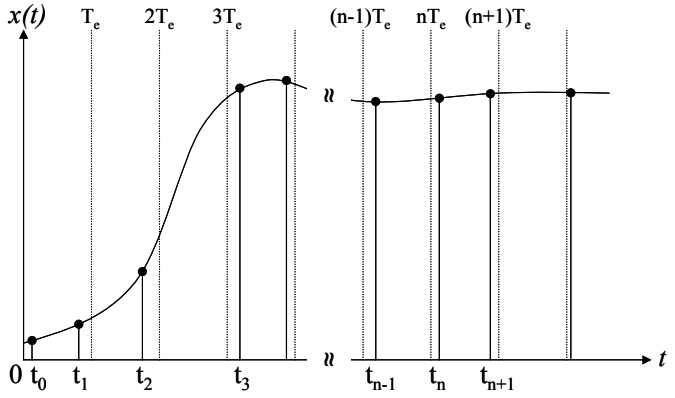


Figure 2 : Echantillonnage uniforme avec un jitter

Il existe plusieurs types de jitter produisant des échantillons irréguliers :

- L'échantillonnage aléatoire uniforme (en anglais *Random Uniform Sampling*) introduisant un biais sur chaque instant d'échantillonnage : $t_n = nT_e + \varepsilon$, où ε est une variable aléatoire.
- L'échantillonnage aléatoire variable (en anglais *Jittered Sampling*) introduisant un jitter sur chaque instant d'échantillonnage : $t_n = nT_e + \varepsilon_n$, où $\{\varepsilon_n\}$ est un ensemble de variables aléatoires indépendantes et identiquement distribuées.
- L'échantillonnage aléatoire cumulatif (en anglais *Additive Random Sampling*) introduisant un jitter qui modifie l'erreur de chaque nouvel échantillon en fonction des erreurs précédentes : $t_n = t_{n-1} + T_e + \varepsilon_n$.

Dans tous les cas, les erreurs étant de moyennes nulles, la fréquence moyenne de l'échantillonnage est égale à F_e. Ainsi bien que ce type d'échantillonnage ait des propriétés

intéressantes, le nombre moyen d'échantillons par unité de temps reste fixé par la fréquence d'échantillonnage. Ce type d'échantillonnage ne nous convient donc pas dans l'optique d'une réduction significative de l'activité d'une chaîne de traitement du signal.

I.1.4.3 Echantillonnage uniforme avec perte d'échantillons

Comme nous l'avons mentionné en introduction de l'échantillonnage non uniforme, il se peut que lors de la transmission, certaines données préalablement échantillonnées régulièrement soient perdues. L'objectif est alors de retrouver les échantillons manquants (figurés par des croix sur la Figure 3) avec de techniques d'interpolation. Nous pouvons citer par exemple les travaux de Marvasti à base de méthodes itératives [Marvasti 2005] et de Ferreira à base de bancs de filtres de reconstruction [Santos *et al.* 2005].

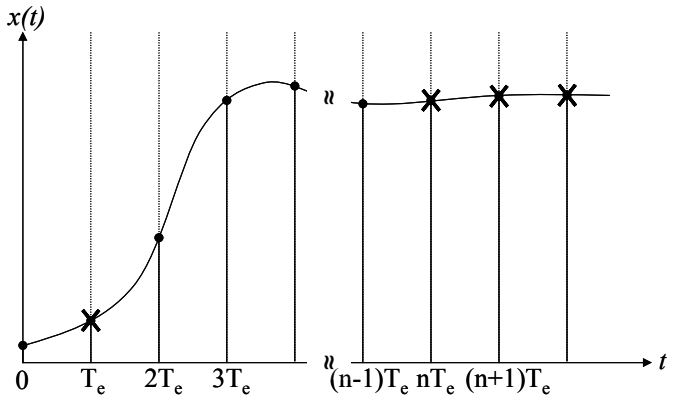


Figure 3 : Echantillonnage uniforme avec perte d'échantillons

Cependant dans certains cas, les échantillons peuvent être volontairement supprimés par un système de décimation. Ce procédé a été utilisé par Fontaine pour compresser les données provenant de signaux de type électrocardiogramme [Fontaine 1999]. Dans un premier temps, le signal est échantillonné régulièrement puis un algorithme ne conserve que les points *utiles* en fonction du signal sous la forme de couples amplitude, intervalle de temps. Cette stratégie permet de réduire l'activité de la chaîne de traitement. Toutefois, en entrée, un système de conversion doit fonctionner en continu rendant cet échantillonnage attractif pour améliorer le stockage des données mais pas pour réduire l'activité électrique du système.

I.1.4.4 Echantillonnage par traversée de niveaux

Le principe de l'échantillonnage par traversée de niveaux est de s'affranchir du signal d'horloge commandant l'échantillonnage dans les systèmes de conversion analogique numérique. En effet, une fréquence d'échantillonnage n'est liée au signal à temps continu que par le théorème de Shannon. Les caractéristiques du signal ne sont pas prises en compte, notamment les zones de faible variation où un échantillonnage uniforme prélève des données redondantes. Mark a donc proposé une méthode où les échantillons sont convertis lorsque le signal croise l'un des niveaux répartis uniformément sur la dynamique du signal d'entrée afin de compresser les données prélevées [Mark *et al.* 1981]. A la différence où l'amplitude du signal est normalement quantifiée, ce type d'échantillonnage introduit une quantification du temps liée à la mesure de la durée séparant deux échantillons successifs. L'intérêt de cet échantillonnage réside dans le fait que l'activité du signal régule automatiquement l'activité du système de conversion. L'échantillonnage par traversée de niveaux est donc très attractif pour réduire l'activité de l'ensemble de la chaîne de traitement du signal et donc sa consommation d'énergie.

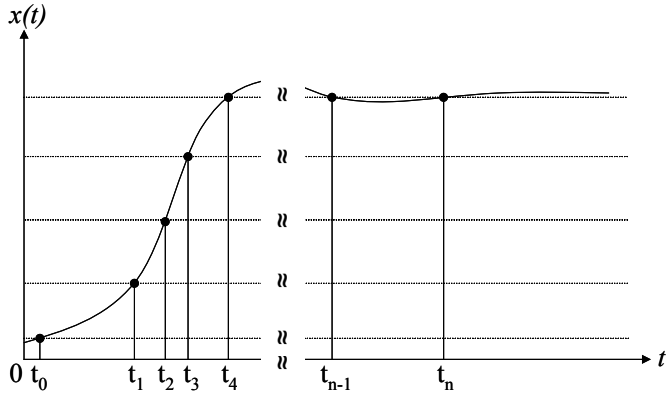


Figure 4 : Echantillonnage non uniforme par traversée de niveaux

Partant de ce principe, un système complet de conversion analogique-numérique basé sur un échantillonnage par traversée de niveaux a été proposé par Sayiner [Sayiner *et al.* 1996]. Il définit ainsi un système composé de trois étages successifs :

1. Un convertisseur à échantillonnage par traversée de niveaux : en étudiant une application donnée, il propose une architecture permettant d'échantillonner non uniformément un signal. Les paramètres de la structure, comme le nombre de niveaux ou la base temporelle servant à la quantification du temps, sont fixés en fonction du rapport signal sur bruit visé.

2. Un interpolateur pour ré-échantillonner le signal régulièrement : il montre alors qu'en interpolant le signal à l'aide de polynômes d'ordre 2, le rapport signal sur bruit n'est pas dégradé. Cela lui permet de se ramener à la théorie classique du traitement du signal : en obtenant en sortie des échantillons uniformément répartis dans le temps, il peut d'une part comparer son système aux structures existantes basées sur l'échantillonnage uniforme et d'autre part utiliser les algorithmes classiques malgré un échantillonnage non uniforme.
3. Un décimateur pour réduire la fréquence d'échantillonnage à la fréquence de Nyquist : il peut ainsi réduire le nombre de point à traiter mais cela lui permet surtout d'augmenter le rapport signal sur bruit en dB de *10log(Facteur-de-décimation)* et par conséquent la résolution effective du convertisseur.

Finalement l'ensemble du convertisseur est équivalent à un système classique utilisant un échantillonnage uniforme. Cependant ce travail est très intéressant car il propose une architecture de conversion bouclée et asservie sur le signal à échantillonner bien que les étages interpolateur et décimateur suppriment l'intérêt de l'échantillonnage par traversée de niveaux dans la réduction de l'activité d'une chaîne de traitement de signal.

I.2 Principes des circuits asynchrones

A la différence d'un circuit synchrone dont le fonctionnement est orchestré par un signal d'horloge global, un circuit asynchrone est un système contrôlé par une multitude de signaux gérant les échanges d'information entre blocs fonctionnels internes. Le fonctionnement est ainsi de type flot de données : chaque bloc attend en effet des données en provenance du bloc précédent, les traite et les envoie au suivant. Le contrôle n'est donc plus global mais localisé : pour traiter une donnée, il faut dans un premier temps attendre un ordre de validation de la part du bloc précédent ; pour ce faire celui-ci doit envoyer une requête. Puis dans un second temps, il faut informer le bloc précédent que la donnée est bien reçue afin que lui-même puisse exécuter une nouvelle tache ; pour cela, il faut lui envoyer un acquittement. Le contrôle utilise donc une signalisation bidirectionnelle ; celle-ci est basée sur un mode de communication dit à poignée de main ou de type requête-acquittement comme le montre la figure suivante.

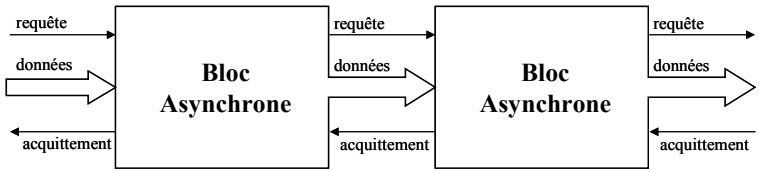


Figure 5 : Schéma fonctionnel d'un circuit asynchrone

I.2.1 Comparaison des circuits synchrones et asynchrones

Depuis des années, la conception des circuits intégrés est réalisée quasi exclusivement à l'aide de fonctions synchrones. En effet, dans la mesure où les communications entre les blocs sont simplifiés par la présence d'une horloge, il n'y a qu'une seule contrainte à respecter pour le concepteur : assurer que le chemin critique ne soit pas plus grand qu'une période d'horloge. Cependant, les tendances actuelles imposent que les circuits soient de taille réduite, faiblement bruités, et surtout peu consommateur d'énergie. Ces nouvelles contraintes de conception dévoilent alors certaines lacunes liées au caractère synchrone d'un circuit. Nous en présentons quelques unes dans cette section.

I.2.1.1 Rapidité de calcul

La rapidité d'un circuit synchrone est définie par son chemin critique; elle est donc limitée par le bloc le plus lent : la fréquence du signal d'horloge doit être réduite en fonction du délai nécessaire pour effectuer cette tâche indépendamment des performances du reste du circuit.

Dans un circuit asynchrone, la tâche réalisée par un module est effectuée dans un temps borné qui dépend directement des données qui le traverse. Ainsi chaque bloc est défini par un temps de latence minimum et temps de latence maximum. Cependant le principe d'un fonctionnement de type flot de donnée est que toutes les sorties peuvent immédiatement être utilisées par le module suivant. Un circuit asynchrone travaillera donc toujours en un temps minimum car les données se propageront systématiquement dans les différents blocs à la vitesse maximum permise par les conditions de fonctionnement.

I.2.1.2 Consommation d'énergie

Dans un circuit synchrone, tous les blocs sont activés en même temps à chaque front d'horloge même ceux dont les données d'entrée n'ont pas évolué. La mémorisation répétée de

l'état des blocs est donc source d'une consommation d'énergie inutile qui croît avec la fréquence de l'horloge.

En revanche, dans un circuit asynchrone, seul le matériel impliqué dans le calcul est activé et consomme de l'énergie. Les blocs qui ne sont pas nécessaires sont naturellement mis en veille par le contrôle local. La consommation du circuit est par conséquent directement liée à l'activité du système donc à l'application.

I.2.1.3 Modularité

La modularité est un aspect fondamental de la conception d'un circuit. Elle permet en effet de concevoir des systèmes complexes entiers à partir de l'utilisation de blocs préexistants. Dans le cas d'une conception synchrone, l'assemblage de modules différents est rendu d'autant plus difficile que tous ont été optimisés séparément pour fonctionner à des fréquences d'horloge différentes. En revanche dans le cas d'une conception asynchrone, la modularité est directement induite par le fonctionnement de type flot de données. En effet, comme le contrôle d'un module est localisé, l'utilisation d'un protocole de communication unique permet de connecter facilement ensemble différents blocs préconçus pour créer un système complexe.

I.2.1.4 Emissions électromagnétiques

La présence d'un signal d'horloge dans les circuits synchrones est source d'émissions électromagnétiques. En effet, chaque front du signal induit des appels de courant répétés et ceux quelle que soit l'activité du circuit. L'augmentation de la fréquence de l'horloge accroît les appels et élargit le spectre des ondes électromagnétiques.

La localisation du contrôle dans un circuit asynchrone permet d'améliorer la répartition temporelle de l'activité électrique car les blocs se sont plus activés sur un front de l'horloge mais par les blocs qui les précèdent. Une meilleure répartition de l'activité électrique réduit les appels de courant et par conséquent la puissance des ondes électromagnétiques émises. Cet avantage des circuits asynchrones a été utilisé à plusieurs reprises dans le cadre d'applications spécifiques. Dhanistha Panyasak [Panyasak 2004] a montré qu'il était possible d'adapter la conception d'un circuit au niveau architecture : en modélisant l'évolution des courants dans le temps, elle propose d'extraire des profils qu'elle réintroduit lors de la description haut niveau du circuit. Cette technique lui permet de montrer qu'un circuit adapté pour générer des courants mieux répartis dans

le temps émet de plus faibles ondes électromagnétiques. Une autre application tirant profit des circuits asynchrones est la cryptographie. En effet, les circuits cryptographiques subissent des attaques par le biais d'analyses qui relient les pics de courant consommé et les données traitées. Comme chaque bloc logique possède une signature électrique particulière en fonction des données qu'il traite, l'analyse des courants par des procédés statistiques ou différentiels donne une représentation de l'activité d'une puce cryptographique et permet de retrouver les données manipulées comme le code secret. Sur le même principe, l'analyse de la signature électromagnétique peut remplacer l'analyse de la signature électrique. La solution proposée par Fraidy Bouesse [Bouesse 2005] est d'utiliser les circuits asynchrones pour équilibrer les chemins de données. Cette technique rend les profils de courant quasiment identiques, immunisant ainsi les circuits de cryptographie contre des attaques liées aux émissions électromagnétiques.

I.2.2 Protocole de communication

Un protocole de communication est un ensemble de règles qui décrit l'échange d'information entre deux parties, un émetteur et un récepteur. Dans le cas d'un circuit asynchrone, chaque bloc doit posséder une interface implémentant le protocole pour générer les signaux de requête et d'acquittement. On désignera comme émetteur un bloc envoyant une requête. Dans la littérature, les différents protocoles sont regroupés en deux grandes catégories : le protocole à deux phases et le protocole à quatre phases.

I.2.2.1 Protocole à deux phases

Le protocole à deux phases établit une communication entre deux blocs asynchrones en deux étapes.

Etape 1 : l'émetteur envoie une requête qui est détectée par le récepteur. Celui-ci reçoit alors les données, les traite et envoie en retour un acquittement.

Etape 2 : l'émetteur détecte l'acquittement qui lui indique que de nouvelles données peuvent être transmises.

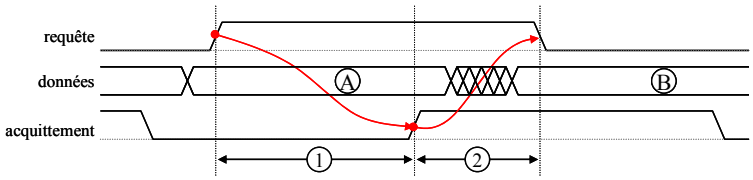


Figure 6 : Protocole à deux phases

Ce protocole est aussi appelé NRZ (Non Retour à Zéro) car comme le montre la Figure 6, les signaux de contrôle n'ont qu'une seule transition par cycle (toutes les deux phases); ils ne sont jamais remis à zéro. Cependant bien que le protocole soit réduit à sa plus simple expression, l'interfaçage est rendu complexe car il doit être capable de détecter des transitions.

I.2.2.2 Protocole à quatre phases

Le protocole à quatre phases est similaire au protocole à deux phases. La principale différence réside dans le fait que chaque signal de contrôle est réinitialisé. Ce protocole est donc également appelé RZ (Retour à Zéro). Les signaux de contrôle ont cette fois-ci deux transitions par cycle (toutes les deux phases); ainsi le protocole est a priori plus complexe que le précédent puisqu'il y a deux fois plus de transitions mais l'interfaçage est simplifié car il suffit de détecter des niveaux logiques et non des transitions. La communication s'effectue de la manière suivante :

Etape 1 : l'émetteur envoie une requête qui est détectée par le récepteur. Celui-ci reçoit alors les données, les traite et envoie en retour un acquittement.

Etape 2 : l'émetteur détecte l'acquittement et réinitialise le signal de requête.

Etape 3 : le récepteur détecte la remise à zéro de la requête et réinitialise l'acquittement.

Etape 4 : l'émetteur détecte la remise à zéro de l'acquittement qui lui indique que de nouvelles données peuvent être transmises.

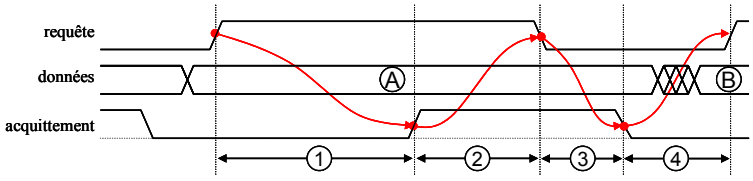
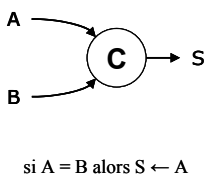


Figure 7 : Protocole à quatre phases

I.2.2.3 La porte de Müller

Dans un circuit asynchrone, l'horloge globale est remplacée par des synchronisations locales. Par conséquent, il est souvent nécessaire de réaliser une synchronisation entre plusieurs signaux. Comme une implémentation utilisant des portes logiques standard est très coûteuse, Müller a proposé une porte « C-element » [Miller 1965] appelée communément aujourd'hui porte de Müller pour réaliser le rendez-vous entre des signaux asynchrones.

La porte de Müller fonctionne sur le principe suivant : si les entrées sont égales, leur valeur est recopiée en sortie. En revanche, si elles ne sont pas égales, la valeur de sortie est mémorisée. Cette porte permet donc d'attendre que toutes les transitions d'entrée soient arrivées pour produire une transition sur le signal de sortie. Le symbole et la table de vérité de la porte sont présentés sur la figure suivante.



A	B	S
0	0	0
0	1	S
1	0	S
1	1	1

Figure 8 : Symbole et table de vérité d'une porte de Müller à 2 entrées

I.2.3 Catégories de circuits asynchrones

Les circuits asynchrones fonctionnent par le biais de communications locales. La synchronisation des tâches doit tenir compte des délais, c'est-à-dire du temps de propagation d'un signal dans une porte logique ou dans une interconnexion. La notion de délai est donc inhérente à la conception asynchrone : plusieurs classes de circuits peuvent être ainsi définies en fonction de la modélisation des délais. Chaque délai est décrit par un modèle temporel :

- Modèle non borné : le retard introduit par une porte logique ou une connexion est positif mais inconnu.
- Modèle borné : le retard est compris dans un intervalle dont seules les limites sont connues.
- Modèle de délai fixe : le retard est une valeur fixe et connue.

Plus le nombre d'hypothèses temporelles utilisées lors de la conception est grand, c'est-à-dire plus les modèles de délai tendent vers des valeurs connues, plus la complexité du circuit est simplifiée. En revanche, le circuit devient moins robuste car chaque erreur de modélisation peut

créer des disfonctionnements. La Figure 9 présente la classification habituelle des circuits asynchrones [Myers 2001]. Ceux-ci vont des circuits de Huffman, faisant le plus d'hypothèses temporelles, aux circuits Insensibles aux Délais, purement asynchrones puisqu'ils n'en font aucune.

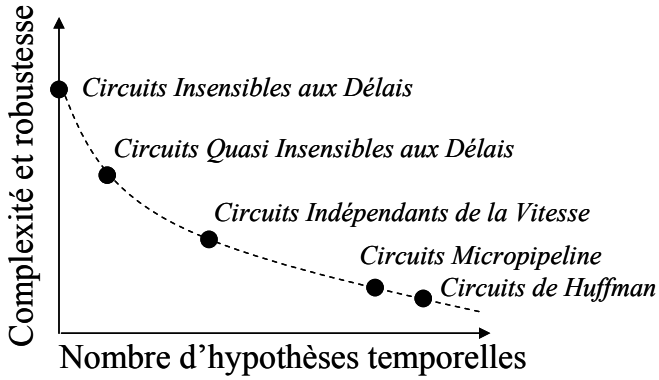


Figure 9 : Catégories de circuits asynchrones

I.2.3.1 Circuits de Huffman

Les circuits de Huffman [Huffman 1964] sont basés sur un modèle de délai borné voire de délai fixé. Lors de la conception, chaque chemin et chaque boucle doivent être finement analysés afin d'établir les modèles temporels car l'implémentation des signaux de contrôle repose sur toutes ces hypothèses. Ces circuits sont donc extrêmement difficiles à caractériser car la moindre erreur de conception dans l'évaluation des délais les rend complètement non fonctionnels.

I.2.3.2 Circuits Micropipelines

Un pipeline est une structure linéaire stockant des données dans des registres et les traitant à l'aide de blocs logiques combinatoires. En ne tenant compte que des registres, un pipeline peut être vu comme une queue dans laquelle transitent les données – la première arrivée étant la première sortie, appelée FIFO en anglais pour *first in first out*. Sur ce principe, Sutherland a proposé [Sutherland 1989] une structure Micropipeline basée sur un modèle de délai borné des blocs de traitement. En effet, les délais des portes et des fils ne sont plus considérés; seuls sont analysés les chemins critiques de chaque bloc de traitement. Cette technique s'apparente à la conception synchrone à la différence qu'il s'agit là des pires cas locaux. Dans cette structure, l'horloge est

remplacée par un contrôle local présenté sur la Figure 10 et composé d'éléments identiques connectés en série et tête-bêche.

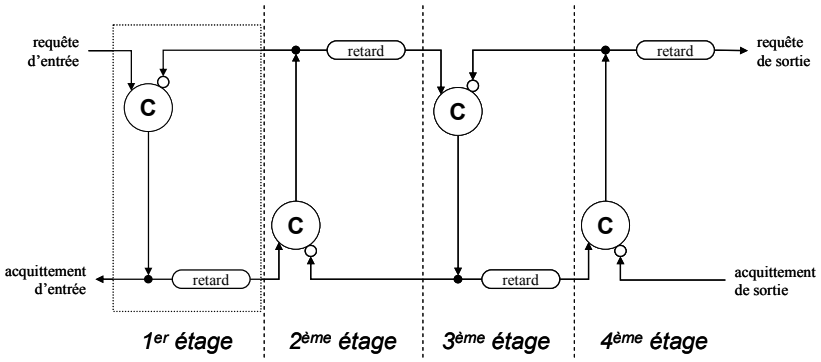


Figure 10 : Contrôle local d'un circuit Micropipeline

Le contrôle fonctionne de la manière suivante : en supposant que tous les signaux sont initialisés à zéro, une transition à '1' du signal de requête d'entrée produit une transition positive du signal d'acquittement d'entrée qui est également propagé au second étage après un délai fixé par le pire cas du premier étage. Le second étage produit alors à son tour une transition à '1' qui est retournée tel un acquittement au premier étage l'autorisant ainsi à traiter une transition à '0' du signal de requête (à cause de la présence de l'inverseur sur la porte de Müller). Parallèlement la transition à '1' du second étage est envoyée au troisième et ainsi de suite. Les transitions avancent donc dans la structure tant qu'elles ne rencontrent pas d'étage occupé. Le protocole utilisé est par conséquent à 2 phases car requêtes et acquittements contrôlent les étages par leurs transitions.

Cette structure est utilisée pour piloter les registres d'un pipeline de traitement. Comme le montre la Figure 11, chaque transition d'une porte de Müller déclenche la capture des données (entrée *c*) en provenance de l'étage précédent. Le registre acquitte l'étage précédent avec le signal *cf* (*capture faite*). Ce signal correspond simplement au signal *c* retardé du temps nécessaire à la mémorisation. Les données étant stockées, l'étage précédent peut commencer un nouveau traitement. A la transition du signal *p* (*passe*) le registre place les données en entrée de la partie combinatoire et génère un signal *pf* (*passe faite*) – le signal *p* retardé – qui permettra sur une requête de commander le registre pour une nouvelle capture.

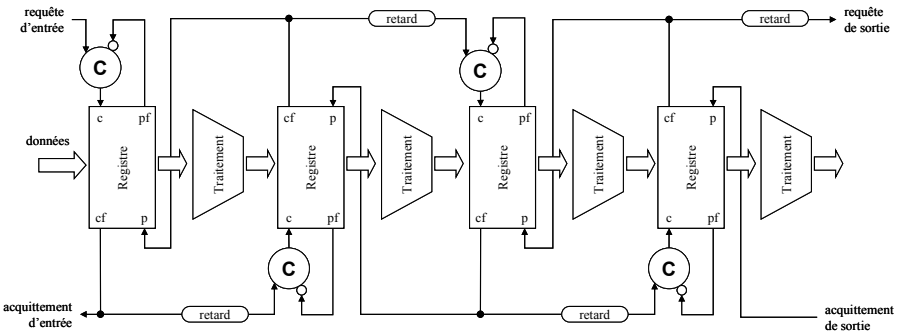


Figure 11 : Circuit Micropipeline avec traitement

Le principal intérêt de ce circuit est que le pipeline FIFO est élastique. Le nombre de données présentes dans la file est variable et les données peuvent donc transiter tant qu'elles ne rencontrent pas d'étage occupé.

I.2.3.3 Circuits Indépendants de la Vitesse

Les circuits Indépendants de la Vitesse (en anglais Speed Independent – SI) utilisent un modèle de délai non borné pour les portes logiques mais supposent que les retards sur les fils sont négligeables [Miller 1965]. Ainsi, dans le cas où un signal est envoyé à plusieurs portes par le biais d'une fourche, une transition est détectée par les portes immédiatement et en même temps. Ces circuits font donc l'hypothèse que les fourches sont isochrones. Or, avec des technologies de plus en plus avancées, le retard induit par les fils n'est plus négligeable; la conception de tels circuits peut alors s'avérer critique.

I.2.3.4 Circuits Insensibles aux Délais

Les circuits Insensibles aux Délais (en anglais Delay Insensitive – DI) utilisent un modèle de délai non borné pour les portes et pour les fils [Clark 1967] [Udding 1986]. Il n'y pas de restriction sur les retards car aucune hypothèse temporelle n'est formulée. Lors de la conception, il faut donc s'assurer que le protocole de communication est respecté et que le circuit fonctionne correctement quels que soient les délais introduit par les portes et par les fils. Ce type de circuit est donc très robuste mais malheureusement très compliqué à concevoir car parmi les portes standard à une sortie, seules les portes de Müller et les inverseurs peuvent être utilisés [Martin 1990].

I.2.3.5 Circuits Quasi Insensibles aux Délais

Les circuits Quasi Insensibles aux Délais (en anglais Quasi Delay Insensitive) utilisent un modèle de délai non borné pour les portes et les fils. Il s'agit d'une sous-classe des circuits DI dans laquelle une hypothèse temporelle a été ajoutée. En effet, en supposant que certaines fourches sont isochrones, Martin a montré [Martin 1991] qu'un circuit Insensible aux Délais devenait réalisable à partir de portes standard – le circuit étant alors *Quasi* Insensible aux Délais. Cette hypothèse rend les circuits QDI très intéressants car elle leur permet d'être conçus à partir des mêmes portes logiques que celles utilisées par les circuits synchrones.

Parallèlement, l'hypothèse de fourche isochrone tend à rapprocher les circuits QDI et SI. Il y a aujourd'hui un compromis pour considérer les deux classes équivalentes. En effet, Hauck a montré [Hauck 1995] qu'une fourche isochrone pouvait être perçue selon les deux classes : soit avec des retards négligeables ε_1 et ε_2 ($\varepsilon_1 \approx \varepsilon_2 \approx 0$) pour les circuits SI, soit avec des retards quelconques pour les circuits QDI. La différence résidant dans le fait que toutes les fourches d'un circuit SI doivent être isochrones et doivent avoir un délai négligeable.

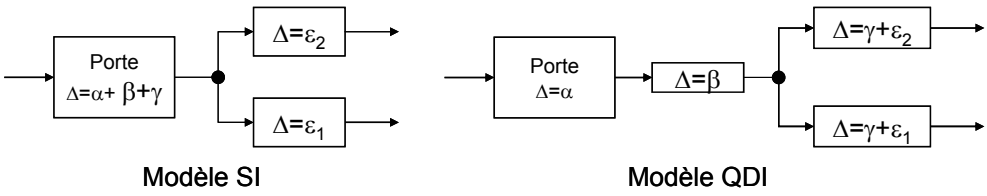


Figure 12 : Equivalence entre les circuits SI et QDI

I.2.4 Convertisseurs analogique numérique asynchrones

Il existe dans la littérature des convertisseurs dits *asynchrones* dans lesquels le principe d'asynchronisme n'est utilisé que pour l'implémentation matérielle. Les concepteurs mettent à profit les avantages des circuits asynchrones pour supprimer certains défauts des convertisseurs. Cependant dans aucun cas, le mode d'échantillonnage n'est remis en cause; l'échantillonnage uniforme est conservé.

Nous pouvons ainsi citer les travaux de Kinniment. Il propose de résoudre les problèmes de métastabilité qui apparaissent en sortie des comparateurs. En effet, si le convertisseur impose un délai fixe au comparateur, celui-ci peut produire une valeur indéterminée pouvant conduire à de graves erreurs en sortie si le bit de poids fort est atteint. Il remplace donc le contrôle des

comparateurs par un système asynchrone qui adapte la mise en forme de la sortie en fonction du temps utilisé par le comparateur. Il conçoit ainsi un convertisseur à approximations successives asynchrone utilisant un seul comparateur dont la tension de référence est mise à jour [Kinniment *et al.* 1998], [Kinniment *et al.* 2000] et un convertisseur flash asynchrone utilisant plusieurs comparateurs avec des tensions de référence fixes [Kinniment *et al.* 1999]. Dans tous les cas, il montre que les performances d'un convertisseur conçu en technologie asynchrone sont améliorées : diminution du bruit, de la métastabilité mais également de la consommation électrique.

I.3 Conclusion : combinaison d'un échantillonnage non uniforme et d'une conception asynchrone

Comme nous venons de le voir dans les sections précédentes, l'utilisation d'un échantillonnage irrégulier peut permettre de réduire l'activité d'une chaîne de traitement du signal en s'adaptant à l'activité du signal. L'échantillonnage doit être non uniforme dans le temps et parmi les différents types mentionnés précédemment, l'échantillonnage par traversée de niveaux semble être le mieux adapté car directement lié au signal. Un exemple de convertisseur a été proposé mais dans le cadre d'une conception synchrone.

Parallèlement, nous savons qu'une conception asynchrone d'un convertisseur améliore ses performances en terme de consommation et de bruit. Cependant, en conservant un échantillonnage uniforme, ce type de convertisseur ne permet pas d'optimiser l'activité de la chaîne de traitement car il ne tient plus compte du signal d'entrée.

La solution que nous proposons est donc de s'affranchir complètement des aspects uniformes de la conversion analogique-numérique pour réaliser une chaîne de traitement dont l'activité ne dépend que du signal en supprimant définitivement le signal d'horloge : la conception matérielle est asynchrone; le séquençage des tâches est contrôlé localement. L'échantillonnage est non uniforme : les points prélevés sont déterminés en fonction du signal lui-même. En combinant conception asynchrone et échantillonnage non uniforme par traversée de niveaux, nous définissons une nouvelle catégorie de chaîne de traitement du signal asynchrone dans son ensemble c'est-à-dire à la fois du point de vue algorithmique et du point de vue matériel. A la base de la chaîne se trouve une nouvelle classe de convertisseurs analogique-numérique purement asynchrones définie par Emmanuel Allier lors de ces travaux de thèse [Allier 2003]. Nous en présentons une synthèse dans la partie suivante.

I.4 Présentation du Convertisseur A/N Asynchrone : le CANA

Le convertisseur A/N Asynchrone fait parti une nouvelle famille de convertisseurs basés sur la combinaison d'une conception asynchrone de la structure matérielle et d'un échantillonnage non uniforme par traversée de niveaux. Pour un convertisseur N bits, 2^N-1 niveaux sont régulièrement répartis sur la dynamique maximale V_{alim}. Nous définissons le quantum, c'est-à-dire la distance entre deux niveaux :

$$q = \frac{V_{alim}}{2^N - 1} \qquad \text{Eq. (15)}$$

Un échantillon doit alors être prélevé chaque fois que le signal d'entrée croise l'un des niveaux.

I.4.1 Principe de fonctionnement du convertisseur

L'architecture retenue pour le convertisseur est une boucle de conversion composée de quatre éléments distincts : un comparateur, un compteur, un convertisseur N/A et un timer. Le système, présenté sur la Figure 13 est un asservissement dont la consigne est le signal analogique à convertir.

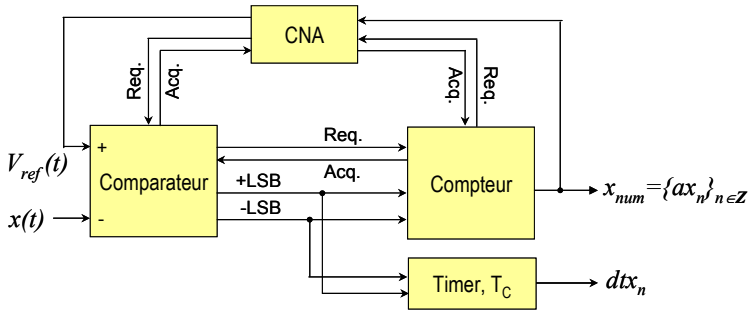


Figure 13 : Boucle de conversion asynchrone

Le principe de fonctionnement de la boucle est le suivant : le signal analogique d'entrée $x(t)$ est converti en un signal numérique x_{num}, lui-même converti par le CNA en un signal de référence $V_{ref}(t)$. Ces deux signaux sont alors comparés : si la différence est plus grande que la moitié d'un quantum, ½q, alors le compteur est incrémenté (*+LSB = '1'*, *-LSB = '0'*) ; si elle est plus petite qu'-½q alors le compteur est décrémenté (*+LSB = '0'*, *-LSB = '1'*) ; dans tous les autres cas, la

sortie reste constante (*+LSB = '0'*, *-LSB = '0'*), il n'y pas d'activité. Comme l'échantillonnage est adapté aux variations du signal d'entrée, donc non uniformément espacé dans le temps, une information temporelle est associée au signal numérique : le temps écoulé depuis l'occurrence précédente est mesuré par le timer. La sortie du convertisseur est ainsi composée de couples (amplitude, intervalle de temps) notés respectivement ax_n, et dtx_n pour spécifier la dépendance au signal d'entrée x. Les instants d'échantillonnage tx_n ne sont pas connus en pratique. Cependant pour simplifier la notation de certains algorithmes, nous pourrons facilement les reconstruire à partir de la relation $tx_n = tx_{n-1} + dtx_n$. Enfin, on peut constater que la structure présentée ne comporte aucun signal d'horloge global. Les blocs sont pilotés par des signaux de contrôle locaux : une requête (*req.*) et un acquittement (*acq.*). Leur synchronisation dépend du protocole de communication choisi lors de l'implémentation pour gérer les échanges.

Le système de conversion est contraint par le signal d'entrée : en effet, les caractéristiques du signal sont limitées par le délai borné δ introduit par la boucle. Ainsi, lorsque la conversion d'un échantillon est commencée (lorsqu'un niveau vient d'être croisé), le signal ne doit pas traverser un autre niveau (déclenchant une nouvelle conversion) avant que le signal de référence n'ait été mis à jour. Ceci limite donc la pente du signal d'entrée :

$$\left|\frac{dx(t)}{dt}\right| \leq \frac{q}{\delta} \qquad \text{Eq. (16)}$$

La relation précédente est appelée condition de poursuite. Si elle n'est pas vérifiée, les échantillons en sortie ne correspondent pas au signal analogique, il y a saturation de la pente.

I.4.2 Quantification du temps et Rapport Signal sur Bruit

Dans un convertisseur A/N asynchrone, un échantillon est prélevé lorsque le signal croise un niveau ; la valeur de l'amplitude est quantifiée et exacte car égale au niveau. Cependant l'instant correspondant au passage reste inconnu ; une mesure temporelle est donc effectuée par un timer induisant une quantification de l'axe des temps et par conséquent une erreur de quantification.

Avec une résolution T_C, le timer compte le nombre d'occurrences qui séparent deux conversions successives. Lorsque le signal traverse un niveau, l'état du timer est figé ; l'instant d'échantillonnage est ainsi connu à une erreur $\delta t \in [0 ; T_C]$ comme l'illustre la Figure 14. Puis, le timer est réinitialisé pour la mesure de l'instant suivant ; la précision d'un intervalle de temps en

sortie du timer dépend donc de la différence des deux erreurs de quantification successives ; l'erreur est donc une comprise dans l'intervalle *[-T_C ; T_C]* :

$$\begin{aligned} dtx_n &= tx_n - tx_{n-1} \\ &= tx_{n,theo} - \delta t_n - tx_{n-1,theo} + \delta t_{n-1} \\ &= dtx_{n,theo} + \delta t_{n-1} - \delta t_n \in \left[dtx_{n,theo} - T_C, dtx_{n,theo} + T_C\right] \end{aligned} \qquad \text{Eq. (17)}$$

La remise à zéro est synchrone sur l'horloge commandant le timer : c'est donc à partir du front d'horloge précédent que le compteur du timer est réinitialisé. Ce mode lui permet de générer des intervalles de temps de moyenne nulle. En reconstruisant les instants d'échantillonnage à partir des intervalles de temps, la quantification introduit un jitter de moyenne nulle (équivalent à un échantillonnage aléatoire variable). Cependant, il est important de noter que si la remise à zéro est asynchrone, c'est-à-dire si la réinitialisation a lieu après un délai donné, les intervalles de temps ne sont plus de moyenne nulle. Les instants d'échantillonnage reconstruits sont alors erronés et chaque nouvelle erreur est accumulée aux précédentes conduisant à une dilatation de l'axe temporel. Par ailleurs, à partir de l'étude de la quantification du temps, il est possible de déterminer le rapport signal sur bruit d'un convertisseur A/N asynchrone. En fonction de la pente instantanée, une erreur en amplitude δx résulte de l'erreur temporelle δt : $\delta x = (dx(t)/dt)\ \delta t$. En considérant les variables $dx(t)/dt$ et δx comme deux variables aléatoires indépendantes, la puissance du bruit de quantification est alors donnée par la relation :

$$P(\delta x) = P\left(\frac{dx}{dt}\right) P(\delta t) \qquad \text{Eq. (18)}$$

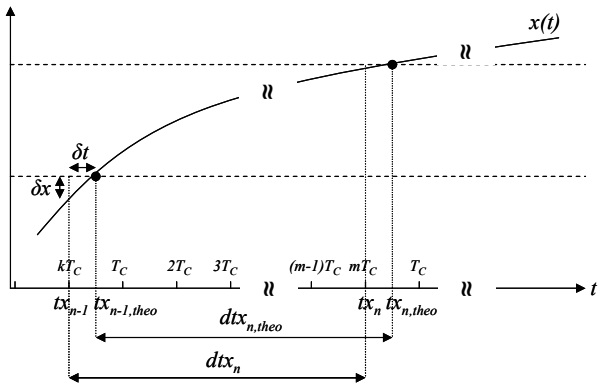


Figure 14 : Quantification du temps induite par le timer lors de la traversée d'un niveau

En supposant que la densité de probabilité de δt est constante dans l'intervalle $[0; T_C]$, la puissance de δt est égale à $P(\delta x) = T_C^2/3$. Le rapport signal sur bruit (RSB) du convertisseur asynchrone est alors défini par la relation :

$$RSB_{dB} = 10\log\left(\frac{P(x)}{P(\delta x)}\right) = 10\log\left(\frac{3P(x)}{P\left(\frac{dx}{dt}\right)}\right) + 20\log\left(\frac{1}{T_C}\right) \qquad \text{Eq. (19)}$$

Le rapport signal sur bruit dépend donc à la fois du signal (et de sa dérivée) et de la résolution du timer. Or, pour une application donnée, c'est-à-dire pour une classe de signaux donnée, le premier terme devient constant. Le rapport ne dépend alors que de la résolution du timer T_C. En revanche, il ne dépend pas de la résolution N du convertisseur, c'est-à-dire du quantum, de la même manière que le RSB d'un convertisseur classique ne dépend pas de la fréquence d'échantillonnage. Pour une résolution matérielle donnée, le RSB du convertisseur peut être réglé en modifiant la période de l'horloge du timer.

Pour déterminer une résolution équivalente avec les convertisseurs classiques, la notion de nombre effectif de bits (*ENOB*) est introduite : elle permet de comparer directement les performances entre les convertisseurs en égalisant les *RSB* dans les cas synchrone et asynchrone puis en remontant à la résolution dans le cas synchrone (selon l'équation (20) pour un signal sinusoïdal par exemple).

$$ENOB = \frac{RSB_{dB} - 1{,}76}{6{,}02} \qquad \text{Eq. (20)}$$

En conclusion on peut remarquer que le convertisseur asynchrone est le cas dual des convertisseurs classiques. Le tableau suivant résume leurs caractéristiques fondamentales.

	Convertisseur A/N classique	CANA
Déclenchement d'une conversion	Horloge	Niveaux
Amplitude	Valeur quantifiée	Valeur exacte
Temps	Valeur exacte	Valeur quantifiée
Dépendance du RSB	Nombre de bits	Résolution du timer
Sortie du convertisseur	Amplitude	(amplitude, intervalle de temps) (± q, intervalle de temps)

Tableau 2 : Caractéristiques comparées des convertisseurs

I.4.3 Implémentation asynchrone du CANA

La structure asynchrone choisie pour implémenter la boucle de conversion est une structure de type micropipeline [Sutherland 1989] à trois étages pour réaliser respectivement le comparateur, le compteur et le convertisseur N/A. Chaque étage est composé de deux parties distinctes : un chemin de donnée et une partie contrôle QDI. Le schéma fonctionnel (sans le timer) est donné Figure 15. La partie contrôle est composée de portes de Muller, d'inverseurs et de retards. Ces derniers correspondent aux délais critiques de chaque étage afin qu'ils n'envoient pas de requête à l'étage suivant avant la fin de l'exécution de leurs calculs respectifs.

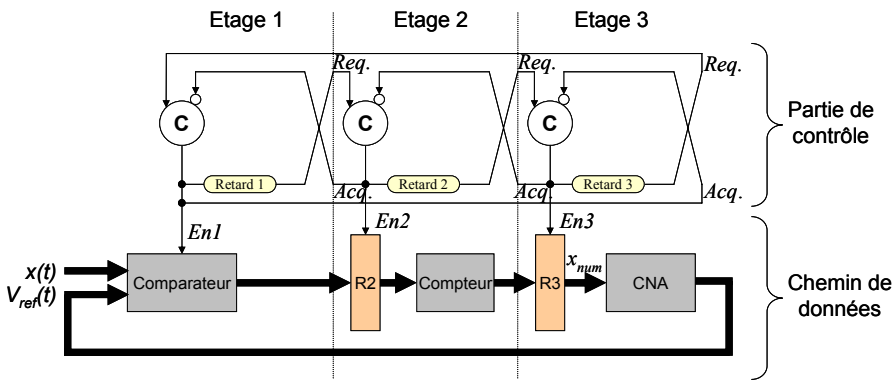


Figure 15 : Schéma fonctionnel asynchrone du CANA

Le fonctionnement de la partie contrôle est le suivant : après initialisation, le 1er étage devient actif (le signal *En1* est validé, *En1 = '1'*); lorsque le signal d'entrée croise un niveau; la sortie du comparateur est stockée et ne peut plus être modifiée avant la fin de la conversion. Une requête est alors transmise au second étage tandis qu'un acquittement est renvoyé au 3ème contrôleur. Celui-ci peut ainsi attendre la prochaine requête du 2ème étage. Les données indiquant le type de croisement (montée ou descente) sont prêtes en entrée du registre *R2*; le 2ème étage est alors activé (*En2 = '1'*). En fonction des valeurs de *R2*, le compteur incrémente (ou décrémente) la valeur x_{num} du registre *R3*. Le 1er étage est acquitté puis après un délai *Retard2*, les données de *R3* sont devenues valides. Une requête est par conséquent envoyée au 3ème contrôleur; le 3ème étage est à son tour activé (*En3 = '1'*). Le signal de référence est mis à jour par la nouvelle valeur numérique via le CNA. Après un délai *Retard3*, la valeur de *Vref* est disponible; une requête remet le comparateur en activité pour une nouvelle conversion (*En1 = '1'*).

A ce jour, un seul convertisseur a été réalisé [Allier *et al.* 2005]. Il s'agit d'un convertisseur 4 bits (15 niveaux) conçu en technologie CMOS 120nm, 1,2V ($q = 40mV$). Le temps de boucle est de 66ns permettant la conversion de signaux de fréquence maximale 160kHz. Les intervalles de temps sont quantifiés sur 12 bits et la résolution du timer peut aller jusqu'à $T_C = 1\mu s$ permettant au CANA d'être équivalent à un convertisseur classique 10 bits. Les tests de consommation ont montré que la puissance *P* du convertisseur est toujours inférieure à 180µW et ce même lorsque le signal d'entrée atteint la fréquence maximale admise par le système. Ceci permet ainsi au CANA d'avoir un facteur de mérite *FoM* (c'est-à-dire un critère global de performance défini par l'équation (21)) deux fois plus grand que les convertisseurs classiques les plus performants.

$$FoM = \frac{2^{ENOB} \cdot f_{max}}{P \cdot Surface} \qquad \text{Eq. (21)}$$

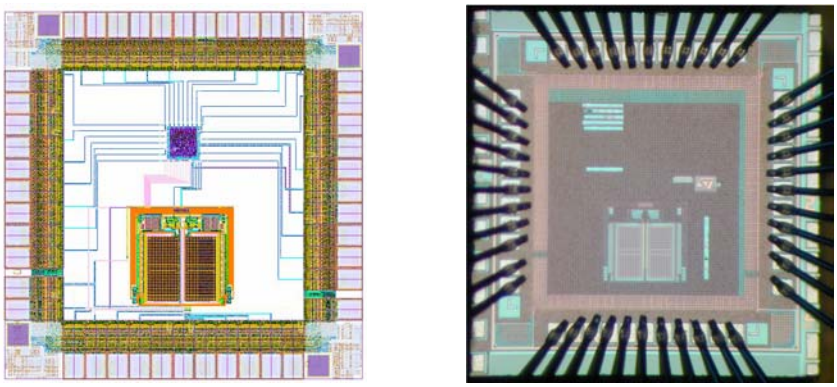


Figure 16 : Layout et photographie du convertisseur CANA

I.4.4 Non linéarité de la conversion asynchrone

Un convertisseur asynchrone échantillonne un signal non uniformément dans le temps. Un échantillonnage irrégulier, défini par un processus aléatoire et stationnaire, génère des instants d'échantillonnage dont les propriétés dépendent exclusivement du processus. Or, l'échantillonnage par traversée de niveaux est lié au signal. Les instants d'échantillonnage dépendent donc des propriétés du signal et non pas d'un processus unique. Le convertisseur est donc non linéaire : les propriétés de l'échantillonnage d'un signal ne peuvent pas être déduites de celles des signaux qui composent le signal. Chaque signal amène une conversion particulière. Nous allons par conséquent étudier la conversion asynchrone de signaux courants, afin de déterminer dans chaque cas les effets de l'échantillonnage sur le plan spectral, notamment les modifications générées sur le spectre du signal analogique.

CHAPITRE II

Etude de l'échantillonnage par traversée de niveaux des signaux périodiques

---

L'échantillonnage par traversée de niveaux étant un processus non linéaire dépendant du signal d'entrée, l'étude formelle de la conversion de signaux usuels doit être réalisée. Le choix des signaux usuels à analyser doit prendre en considération leurs variations car c'est en fonction de leur évolution que l'échantillonnage est commandé. Ainsi, nous proposons dans ce chapitre d'étudier la conversion de signaux périodiques car leurs dérivées ont la propriété d'être également périodiques. Nous faisons donc l'hypothèse de signaux non bruités pour rester dans le cadre d'une analyse théorique de l'échantillonnage (le bruit provenant généralement du capteur).

Dans un premier temps, une étude de cas est réalisée à partir d'un signal sinusoïdal car il permet facilement d'analyser les instants d'échantillonnage et les intervalles de temps en fonction de ses paramètres – fréquence fondamentale, amplitude crête et amplitude moyenne – pour un convertisseur donné. Puis, dans un second temps une généralisation à l'ensemble des signaux périodiques est proposée.

Dans chaque cas, le schéma d'échantillonnage, i.e. la modification effectué sur le spectre du signal analogique par l'échantillonnage, est caractérisé en montrant comment le spectre du signal à temps discret est replié. Ce résultat nous permettra de prouver que la GDFT, qui calcule le spectre du signal échantillonné, n'est pas l'applicable pour analyser le signal analogique dans le cadre d'un échantillonnage par traversée de niveaux et ce quel que soit le signal considéré (ce résultat restera valable dans le chapitre III concernant les signaux non périodiques).

II.1 Etude de cas : échantillonnage d'un signal sinusoïdal

L'échantillonnage par traversée de niveaux est contrôlé par les variations du signal d'entrée. Il dépend donc aussi du caractère répétitif du signal car la dérivée d'une fonction périodique est aussi périodique. Nous allons nous intéresser à un cas d'école : une fonction sinusoïdale. Soit *x(t)* un signal analogique de valeur moyenne a_M, d'amplitude crête a_o, et de période $T_o = 1/f_o$:

$$x(t) = a_M + a_o \sin(2\pi f_o t) \qquad \text{Eq. (22)}$$

L'échantillonnage étant non uniforme, les intervalles de temps évoluent en fonction des variations de la sinusoïde. Il est possible dans ce cas précis de les étudier analytiquement en inversant la fonction étant donnée qu'elle est bijective dans l'intervalle $[-T_o/4 ; T_o/4]$.

II.1.1 Etude des intervalles de temps

Pour un niveau donné a_n parmi les N niveaux possibles, l'amplitude du signal permet de calculer les instants d'échantillonnage tx_i à partir de la phase $\theta x_i = 2\pi f_o tx_i$:

$$a_M + a_o \sin(\theta x_i) = a_n \qquad \text{Eq. (23)}$$

$$\begin{aligned} \arcsin\left((a_n - a_M)/a_o\right) &= \theta x_i + 2k\pi \qquad \theta x_i \in \left[-\frac{\pi}{2};\frac{\pi}{2}\right] \\ &= \pi - \theta x_i + 2k\pi \quad \theta x_i \in \left[\frac{\pi}{2};\frac{3\pi}{2}\right] \end{aligned} \qquad \text{Eq. (24)}$$

$$\begin{aligned} tx_i &= \frac{\arcsin\left((a_n - a_M)/a_o\right)}{2\pi f_0} + kT_o \qquad k \in \mathbf{N},\ tx_i \in \left[\frac{-T_o}{4};\frac{T_o}{4}\right] \\ &= \left(\frac{2k+1}{2}\right)T_o - \frac{\arcsin\left((a_n - a_M)/a_o\right)}{2\pi f_o} \quad k \in \mathbf{N},\ tx_i \in \left[\frac{T_o}{4};\frac{3T_o}{4}\right] \end{aligned} \qquad \text{Eq. (25)}$$

Les instants d'échantillonnage sont périodiques de période T_o, la période fondamentale du signal analogique. Nous pouvons noter, pour faire le parallèle avec l'échantillonnage régulier, qu'un signal périodique donne systématiquement une série de points non uniforme mais périodique. Dans le cas classique, la périodicité dépend du rapport entre la fréquence d'échantillonnage et la fréquence fondamentale.

L'intervalle de temps entre deux points successifs se calcule par le délai séparant deux instants d'échantillonnage. Ainsi, plusieurs cas se distinguent en fonction de la position des instants dans une période du signal.

II.1.1.1 Intervalles de temps des variations croissantes

Lors d'une variation croissante du signal, c'est-à-dire dans l'intervalle $[-T_o/4; T_o/4]$, le signal croise successivement le niveau a_{n-1} à l'instant tx_{i-1} puis le niveau $a_n = a_{n-1} + q$ à l'instant tx_i. L'intervalle de temps dtx_i entre les deux passages est donc déterminé par la relation :

$$\begin{aligned} dtx_i &= tx_i - tx_{i-1} \\ &= \frac{\arcsin\left(\left(a_n - a_M\right)/a_o\right)}{2\pi f_o} + kT_o - \frac{\arcsin\left(\left(a_{n-1} - a_M\right)/a_o\right)}{2\pi f_o} - kT_o \\ &= \frac{T_o}{2\pi}\left(\arcsin\left(\left(a_n - a_M\right)/a_o\right) - \arcsin\left(\left(a_{n-1} - a_M\right)/a_o\right)\right) \end{aligned} \qquad \text{Eq. (26)}$$

Pour des niveaux a_n et a_{n-1} donnés, l'intervalle de temps est donc unique car indépendant du paramètre k. Il est de plus périodique de période T_o car tx_i et tx_{i-1} sont eux-mêmes périodiques de période T_o.

Par ailleurs, il est important de noter que la valeur des intervalles de temps est non linéaire avec les paramètres liés à l'amplitude (a_M et a_o) et qu'au contraire, la valeur est linéaire avec la période T_o du signal.

II.1.1.2 Intervalles de temps des variations décroissantes

Lors d'une variation décroissante du signal, c'est-à-dire dans l'intervalle $[T_o/4 ; 3T_o/4]$, le signal croise cette fois-ci le niveau a_n à l'instant tx_{l-1} puis le niveau a_{n-1} à l'instant tx_l. L'intervalle de temps dtx_l est donc déterminé par :

$$\begin{aligned} dtx_l &= tx_l - tx_{l-1} \\ &= \left(\frac{2k+1}{2}\right)T_o - \frac{\arcsin\left(\left(a_{n-1} - a_M\right)/a_o\right)}{2\pi f_o} - \left(\frac{2k+1}{2}\right)T_o + \frac{\arcsin\left(\left(a_n - a_M\right)/a_o\right)}{2\pi f_o} \\ &= \frac{T_o}{2\pi}\left(\arcsin\left(\left(a_n - a_M\right)/a_o\right) - \arcsin\left(\left(a_{n-1} - a_M\right)/a_o\right)\right) = dtx_i \end{aligned} \qquad \text{Eq. (27)}$$

Le temps pour passer du niveau *n* au niveau *n-1* est donc égal au temps pour passer du niveau a_{n-1} au niveau a_n (partie gauche de la Figure 17). La variation des intervalles de temps est par conséquent symétrique par rapport aux crêtes du signal analogique. Toutefois, dans le plan (amplitude, intervalle de temps), l'échantillonnage provoque un cycle hystérésis car deux intervalles de temps égaux correspondent à deux niveaux décalés d'un quantum entre la partie croissante et la partie décroissante du signal (partie droite de la Figure 17).

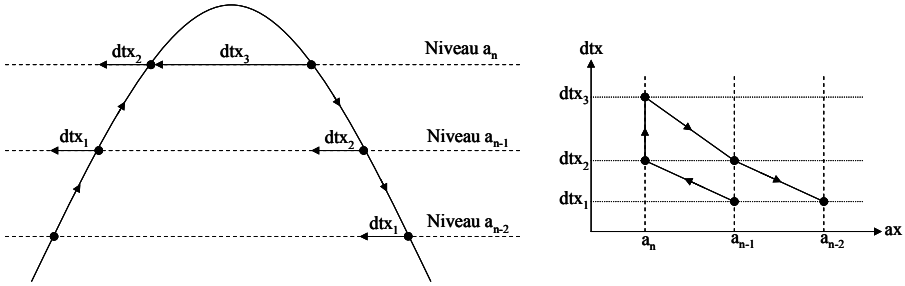


Figure 17 : Cycle hystérésis des intervalles de temps autour d'une crête

II.1.1.3 Intervalles de temps des crêtes du signal

Lors d'une crête positive, le signal passe deux fois successivement par le niveau a_{SUP} aux instants tx_{i-1} inclus dans l'intervalle $[-T_o/4; T_o/4]$ et tx_i contenu dans l'intervalle $[T_o/4 ; 3T_o/4]$. L'intervalle de temps dtx_{SUP} est alors donné par :

$$\begin{aligned} dtx_{SUP} &= tx_i - tx_{i-1} \\ &= \left(\frac{2k+1}{2}\right)T_o - \frac{\arcsin\left((a_{SUP}-a_M)/a_o\right)}{2\pi f_o} - \frac{\arcsin\left((a_{SUP}-a_M)/a_o\right)}{2\pi f_o} - kT_o \quad \text{Eq. (28)} \\ &= T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{a_{SUP}-a_M}{a_o}\right)\right) \end{aligned}$$

En revanche lors d'une crête négative, le signal traverse le même niveau a_{INF} aux instants tx_{j-1} de l'intervalle $[T_o/4 ; 3T_o/4]$ et tx_j de l'intervalle $[-T_o/4; T_o/4]$. L'intervalle de temps dtx_{INF} est alors déterminé par la relation :

$$\begin{aligned} dtx_{INF} &= tx_j - tx_{j-1} \\ &= \frac{\arcsin\left(\left(a_{INF} - a_M\right)/a_o\right)}{2\pi f_o} + (k+1)T_o - \left(\frac{2k+1}{2}\right)T_o + \frac{\arcsin\left(\left(a_{INF} - a_M\right)/a_o\right)}{2\pi f_o} \\ &= T_o\left(\frac{1}{2} + \frac{1}{\pi}\arcsin\left(\frac{a_{INF} - a_M}{a_o}\right)\right) \end{aligned} \quad \text{Eq. (29)}$$

Ainsi, nous pouvons noter que les intervalles de temps dtx_{SUP} et dtx_{INF} sont eux aussi uniques et périodique de période T_o. Par ailleurs, comme ils dépendent de la valeur des niveaux a_{SUP} et a_{INF}, ils sont bornés dans l'intervalle suivant :

$$dtx_i, dtx_j \in \left[0; T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{q}{a_o}\right)\right)\right[\quad \text{Eq. (30)}$$

II.1.2 Facteur de symétrie

Les intervalles de temps, quelle que soit leur position, dépendent de la fonction *arcsin( )*. Or cette fonction est impaire, ce qui implique que deux intervalles de temps peuvent être égaux pendant la même variation (entre deux crêtes). En effet, s'il existe un ensemble de niveaux régulièrement espacés sur lequel le signal analogique est centré, c'est-à-dire s'il existe un couple de valeurs distinctes *(n, m)* tel que :

$$\left|a_n - a_M\right| = \left|a_M - a_m\right| \quad \text{Eq. (31)}$$

alors le temps nécessaire au signal pour passer du niveau *n-1* au niveau *n* est le même que celui pour passer du niveau *m* au niveau *m+1*. Cette condition dépend uniquement, pour une grille donnée de niveaux, de la valeur moyenne du signal a_M puisque l'équation (31) peut être simplifiée par :

$$a_M = \frac{a_n + a_m}{2} \quad \text{Eq. (32)}$$

a_M est donc également la valeur moyenne de deux niveaux, ce qui implique que a_M correspond soit à un niveau, soit au centre de deux niveaux que nous appellerons *inter-niveau*.

Bien qu'un signal sinusoïdal ne soit pas, a priori, un cas favorable pour l'échantillonnage par traversée de niveaux, il servira de signal de test pour l'étude des traitements numériques. Afin

de se libérer de toutes contraintes liées à la valeur moyenne du signal, donc à la symétrie de l'échantillonnage, il convient d'étudier son influence. Pour ce faire, nous utiliserons les intervalles de temps dont les valeurs sont liées à la symétrie.

En réalisant l'échantillonnage par traversée de niveaux sur une période d'un signal sinusoïdal d'amplitude crête pour différentes valeur moyenne du signal, nous pouvons observer sur la Figure 18 l'évolution temporelle des intervalles de temps en fonction de la valeur moyenne. Nous pouvons noter que seuls les intervalles de temps des points correspondant aux crêtes (respectivement aux instants 0.3s et 0.8s sur la figure) évoluent significativement d'une valeur moyenne à l'autre.

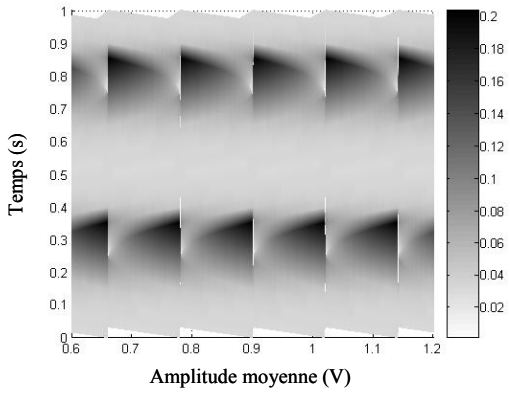


Figure 18 : Evolution temporelle des intervalles de temps en fonction de la valeur moyenne signal

Nous décidons donc d'étudier la symétrie de l'échantillonnage uniquement à travers les valeurs des intervalles de temps des crêtes supérieure et inférieure, respectivement dtx_{SUP} et dtx_{INF}. Nous définissons ainsi le facteur de symétrie de l'échantillonnage par traversée de niveaux du signal x comme le rapport entre l'écart des deux intervalles et leur valeur moyenne :

$$FSx = 2\frac{\left|dtx_{SUP} - dtx_{INF}\right|}{dtx_{SUP} + dtx_{INF}} \qquad \text{Eq. (33)}$$

En fonction de la valeur moyenne du signal, FSx varie entre 0 où l'échantillonnage est symétrique et 2 où l'échantillonnage est dissymétrique.

Pour étudier complètement le facteur de symétrie, nous introduisons un paramètre supplémentaire noté ε correspondant à l'écart maximum entre la crête du signal et le dernier niveau

croisant le signal, lorsque l'autre crête est elle-même tangente à un niveau. Il peut être défini simplement par la relation suivante :

$$\varepsilon = 2a_o - q\left\lfloor \frac{2a_o}{q} \right\rfloor, \qquad \text{Eq. (34)}$$

où $\lfloor \theta \rfloor$ représente la partie entière de θ. Le paramètre ε dépend donc uniquement de l'amplitude crête du signal et de la valeur du quantum, c'est-à-dire du convertisseur. Il peut prendre n'importe quelle valeur comprise dans l'intervalle *[0 ; q[*, les extrêmes correspondant respectivement au cas où les deux crêtes sont tangentes d'un niveau et au cas où une seule est tangente tandis que l'autre tend à l'être aussi (d'où la présence de l'exclusion car sinon les bornes correspondraient au même cas).

En faisant une représentation *tête-bêche* des crêtes en fonction de la valeur moyenne du signal, c'est-à-dire en les comparant sans se préoccuper du retard de $T_o/2$ qui les sépare, nous pouvons dégager deux situations qui sont illustrées sur la Figure 19 :

- Autour d'un niveau
- Autour d'un inter-niveau

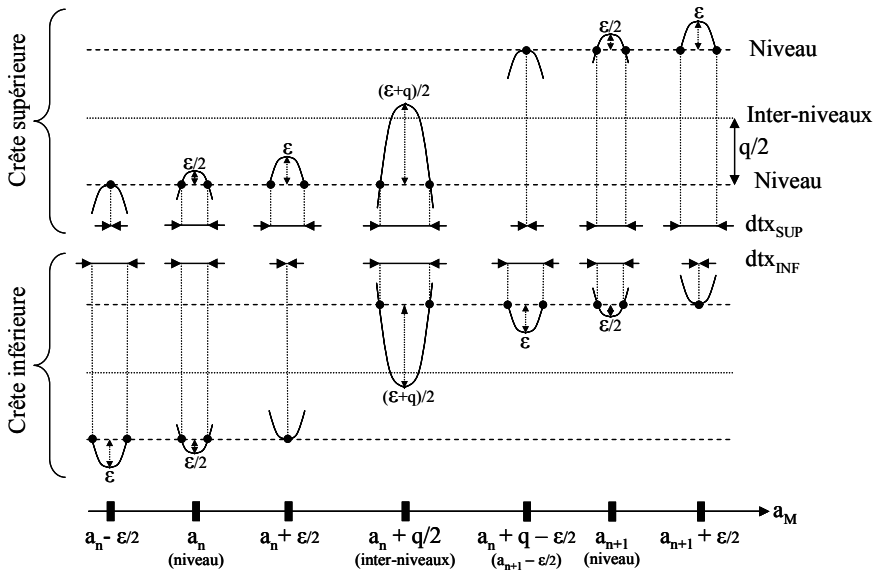


Figure 19 : Représentation *tête-bêche* des deux crêtes d'un signal sinusoïdal en fonction de sa valeur moyenne

II.1.2.1 Evolution du facteur de symétrie autour d'un niveau

Lorsque la valeur moyenne du signal se situe sur un niveau, l'échantillonnage est symétrique, impliquant d'une part que les intervalles de temps des deux crêtes sont égaux et d'autre part que les crêtes dépassent (en valeur absolue) de la même longueur $\varepsilon/2$ leurs niveaux respectifs a_{SUP} et a_{INF}, comme le montre les parties gauche et droite de la Figure 19. On peut observer alors qu'en augmentant la valeur moyenne de $\pm\varepsilon/2$, l'une des crêtes devient tangente au niveau, rendant l'échantillonnage dissymétrique.

Pour étudier l'évolution du facteur de symétrie, il faut donc faire varier la valeur moyenne du signal autour d'une position stable (autour d'un niveau $a_n = a_{M,0}$) pour laquelle l'échantillonnage est symétrique :

$$a_M(\Delta) = a_{M,0} + \Delta \qquad \text{Eq. (35)}$$

où Δ représente une variation de la valeur moyenne du signal autour du niveau $a_n = a_{M,0}$. D'après la Figure 19, Δ appartient à l'intervalle $[-\varepsilon/2; \varepsilon/2]$.

Les intervalles de temps calculés à partir des équations (28) et (29) de la section II.1.1.3 sont maintenant définis en fonction de la variation Δ :

$$\begin{cases} dtx_{SUP}(\Delta) = T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{a_{SUP} - a_M(\Delta)}{a_o}\right)\right) \\ dtx_{INF}(\Delta) = T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{a_M(\Delta) - a_{INF}}{a_o}\right)\right) \end{cases} \qquad \text{Eq. (36)}$$

Pour $\Delta = 0$, nous retrouvons les intervalles de temps pour une position symétrique autour d'un niveau :

$$\begin{cases} dtx_{SUP,0} = T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{a_o - \varepsilon/2}{a_o}\right)\right) \\ dtx_{INF,0} = T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{a_o - \varepsilon/2}{a_o}\right)\right) = dtx_{SUP,0} \end{cases} \qquad \text{Eq. (37)}$$

Pour de petites variations de Δ, i.e. $\Delta \ll \varepsilon$, les niveaux que traverse le signal lors de l'échantillonnage restent inchangés :

$$\begin{cases} a_{SUP} = a_{M,0} + a_o - \dfrac{\varepsilon}{2} \\ a_{INF} = a_{M,0} - a_o + \dfrac{\varepsilon}{2} \end{cases} \qquad \text{Eq. (38)}$$

Ainsi à partir de l'équation (36), en combinant les équations (35) et (38), les intervalles de temps se calculent à partir des mêmes niveaux :

$$\begin{cases} dtx_{SUP}(\Delta) = T_o\left(\dfrac{1}{2} + \dfrac{1}{\pi}\operatorname{asin}\left(\dfrac{\Delta}{a_o} - \dfrac{a_o - \varepsilon/2}{a_o}\right)\right) \\ dtx_{INF}(\Delta) = T_o\left(\dfrac{1}{2} - \dfrac{1}{\pi}\operatorname{asin}\left(\dfrac{\Delta}{a_o} + \dfrac{a_o - \varepsilon/2}{a_o}\right)\right) \end{cases} \qquad \text{Eq. (39)}$$

L'équation (39) montre que les intervalles de temps évoluent de façon non linéaire en fonction de Δ (loi en arc sinus). De la même manière, étant donné que le facteur de symétrie dépend des intervalles de temps, nous supposons que l'évolution approche une fonction en arc sinus. Pour le vérifier, nous proposons d'effectuer un développement limité de la fonction arc sinus afin d'étudier l'évolution sur une courte plage de Δ. En utilisant la formule de Taylor (rappelée par l'équation (40)) à l'ordre 1, nous pouvons approcher la fonction arc sinus par une fonction affine :

$$\underset{X\to\alpha}{f(X)} = \sum_{n=0}^{\infty} \frac{f^{(n)}(\alpha)}{n!}(X-\alpha)^n \qquad \text{Eq. (40)}$$

$$\underset{X\to\alpha}{\operatorname{asin}(X)} = \arcsin(\alpha) + \frac{(X-\alpha)}{\sqrt{1-\alpha^2}} + o(X-\alpha) \qquad \text{Eq. (41)}$$

La fonction $o(X-\alpha)$ est définie pour tendre vers 0 quand X tend vers α. En posant le changement de variables suivant :

$$\begin{cases} \alpha = \dfrac{a_o - \varepsilon/2}{a_o} \\ X = \dfrac{\Delta}{a_o} \end{cases}, \qquad \text{Eq. (42)}$$

nous pouvons déduire la valeur des intervalles de temps en fonction de Δ :

$$\begin{cases} dtx_{SUP}(\Delta) \approx dtx_{SUP,0} + \dfrac{2T_o}{\pi} \dfrac{\Delta}{\sqrt{4a_o\varepsilon - \varepsilon^2}} \\ dtx_{INF}(\Delta) \approx dtx_{INF,0} - \dfrac{2T_o}{\pi} \dfrac{\Delta}{\sqrt{4a_o\varepsilon - \varepsilon^2}} \end{cases} \qquad \text{Eq. (43)}$$

Notons que l'équation (43) n'est valable que pour tout ε non nul. Or, d'après ce qui a été vu ci-dessus, si ε est nul, Δ est forcement nul aussi puisque Δ appartient à l'intervalle $[-\varepsilon/2; \varepsilon/2]$. Ceci lève donc l'ambiguïté car il n'y a plus d'intérêt d'étudier les évolutions des intervalles de temps en fonction de la valeur moyenne dans une plage de variation nulle.

Par ailleurs, nous remarquons que l'accroissement des intervalles de temps en fonction de Δ est le même (en valeur absolue) pour les deux crêtes. Ce résultat est tout à fait normal puisque les crêtes ont la même forme. Nous pouvons donc en déduire l'évolution du facteur de symétrie pour de petites variations :

$$FSx(\Delta) \approx \frac{4}{\pi\sqrt{4a_o\varepsilon - \varepsilon^2}} \frac{T_o}{dtx_{SUP,0}} |\Delta| \qquad \text{Eq. (44)}$$

Pour des grandes valeurs de Δ, i.e. Δ tendant vers $\varepsilon/2$, nous savons que le facteur de symétrie tend vers 2. Nous pouvons en déduire que l'évolution est modélisable par un arc sinus défini par :

$$FSx(\Delta) \approx \frac{4}{\pi} \arcsin\left(\frac{2}{\varepsilon}\left|a_{M,0} - \Delta\right|\right) + \xi(\Delta) \qquad \text{Eq. (45)}$$

où $\xi(\Delta)$ représente l'erreur de modélisation.

Entre des valeurs de Δ intermédiaires, l'évolution du facteur de symétrie reste inconnue mais les modèles décrits par les équations (43) et (44) nous permettront, lors de simulations, d'analyser l'allure du facteur de symétrie.

II.1.2.2 Evolution du facteur de symétrie autour d'un inter-niveau

Le raisonnement à suivre pour étudier l'évolution du facteur de symétrie autour d'un inter-niveau est identique à celui tenu précédemment autour d'un niveau. Dans cette configuration, nous observons au centre de la Figure 19 que les crêtes dépassent (en valeur absolue) cette fois-ci d'une

hauteur $(\varepsilon+q)/2$ leurs niveaux respectifs a_{SUP} et a_{INF}. Ainsi, en augmentant la valeur moyenne de $\pm(q-\varepsilon)/2$, l'une des crêtes devient tangente d'un niveau, rendant l'échantillonnage dissymétrique.

Pour étudier l'évolution du facteur de symétrie, il faut donc à nouveau faire varier la valeur moyenne du signal autour d'une position stable (cette fois ci autour d'un inter-niveau $a_{M,0} = a_n+q/2$) pour laquelle l'échantillonnage est symétrique. La relation (35) est donc conservée ; les variations de la valeur moyenne Δ appartiennent cependant cette fois-ci à l'intervalle $[-(q-\varepsilon)/2 ; (q-\varepsilon)/2]$.

Pour $\Delta = 0$, nous calculons les intervalles de temps pour une position symétrique autour d'un inter-niveau :

$$dtx_{SUP,0} = dtx_{INF,0} = T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{a_o - (\varepsilon+q)/2}{a_o}\right)\right) \qquad \text{Eq. (46)}$$

Puis pour de petites variations de Δ, i.e. $\Delta \ll \varepsilon$, comme les niveaux que traverse le signal lors de l'échantillonnage restent inchangés, les intervalles de temps en fonction de Δ s'écrivent :

$$\begin{cases} dtx_{SUP}(\Delta) = T_o\left(\frac{1}{2} + \frac{1}{\pi}\arcsin\left(\frac{\Delta}{a_o} - \frac{a_o - (\varepsilon+q)/2}{a_o}\right)\right) \\ dtx_{INF}(\Delta) = T_o\left(\frac{1}{2} - \frac{1}{\pi}\arcsin\left(\frac{\Delta}{a_o} + \frac{a_o - (\varepsilon+q)/2}{a_o}\right)\right) \end{cases} \qquad \text{Eq. (47)}$$

A l'aide du développement limité de la fonction arc sinus à l'ordre 1 (équation (41)) et en posant un nouveau changement de variable défini par la relation suivante :

$$\begin{cases} \alpha = \dfrac{a_o - (\varepsilon+q)/2}{a_o} \\ X = \dfrac{\Delta}{a_o} \end{cases}, \qquad \text{Eq. (48)}$$

nous pouvons déduire d'une part l'évolution des intervalles de temps des crêtes :

$$\begin{cases} dtx_{SUP}(\Delta) \approx dtx_{SUP,0} + \dfrac{2T_o}{\pi} \dfrac{\Delta}{\sqrt{4a_o(\varepsilon+q)-(\varepsilon+q)^2}} \\ dtx_{INF}(\Delta) \approx dtx_{INF,0} - \dfrac{2T_o}{\pi} \dfrac{\Delta}{\sqrt{4a_o(\varepsilon+q)-(\varepsilon+q)^2}} \end{cases} \qquad \text{Eq. (49)}$$

et d'autre part l'évolution du facteur de symétrie autour d'un inter-niveau :

$$FSx(\Delta) = \frac{4}{\pi\sqrt{4a_o(\varepsilon+q)-(\varepsilon+q)^2}} \frac{T_o}{dtx_{SUP,0}} \Delta \qquad \text{Eq. (50)}$$

Il est important de noter que l'accroissement du facteur de symétrie est dans ce cas plus petit que celui du cas précédent et ce quelle que soit la valeur de ε. De plus, il n'est pas possible d'étudier l'évolution du facteur pour des variations de Δ plus grandes. En effet, étant donné que la hauteur de la crête dans la position symétrique est de $(q+\varepsilon)/2$, la position dissymétrique aurait dû être obtenue en augment la valeur moyenne de cette même grandeur. Or *entre-amplitudes* (à défaut de pouvoir écrire *entre-temps*) est apparue une dissymétrie, précisément après $(q-\varepsilon)/2$. L'évolution *naturelle* du facteur de symétrie subit donc une discontinuité en ce point. Sans celle-ci, l'évolution peut être modélisée à une erreur $\xi(\Delta)$ près par la relation suivante :

$$FSx(\Delta) = \frac{4}{\pi}\operatorname{asin}\left(\frac{2}{\varepsilon+q}\left|a_{M,0}-\Delta\right|\right) + \xi(\Delta) \qquad \text{Eq. (51)}$$

Cependant grâce à la discontinuité, nous nous affranchissons de cette incertitude car l'évolution reste dans la zone linéaire décrite par l'équation (50). La limite de cette hypothèse est atteinte lorsque ε tend vers 0, c'est-à-dire lorsque la discontinuité ($(q-\varepsilon)/2$) et l'amplitude correspondant à la dissymétrie naturelle ($(q+\varepsilon)/2$) se rejoignent en $q/2$. En effet dans ce cas précis, la pente est la plus grande et le facteur de symétrie doit diverger vers 2 en $\Delta = q/2$. L'évolution du facteur de symétrie autour d'un inter-niveau est alors caractérisée pour de petites variations Δ par l'équation (50) puis pour de grandes variations par l'équation (51) comme nous l'avons vu dans la section précédente pour la symétrie autour d'un niveau.

II.1.2.3 Evolution du facteur de symétrie dans son ensemble

En regroupant ce qui a été vu dans les deux sections précédentes et à l'aide de la Figure 19, nous pouvons en déduire l'évolution du facteur de symétrie dans son ensemble, c'est-à-dire en en

augmentant la valeur de moyenne de telle sorte que le facteur passe successivement de zones symétriques, soit autour d'un niveau soit autour d'un inter-niveau, à des zones dissymétriques.

Pour illustrer simplement nos propos, nous dessinerons toutes les variations non linéaires par des droites.

Sur la Figure 20, nous pouvons observer que l'intervalle de temps d'une crête est nul lorsqu'elle est tangente à un niveau. Puis, il augmente pour atteindre son maximum juste avant que la crête ne soit tangente au niveau suivant. Parallèlement, l'intervalle de temps de la crête opposée suit une trajectoire inverse tout en étant décalé d'un offset ε. La partie supérieure de la Figure 20 qui résume ce qui vient d'être écrit montre que les intervalles de temps se croisent à tous les niveaux et inter-niveaux, et qu'ils sont le plus distant l'un de l'autre à chaque dissymétrie. Le facteur de symétrie illustré sur la partie inférieure de la Figure 20 oscille donc entre 0 et 2, i.e. entre zones symétriques et dissymétriques. De plus, il varie de manière plus rapide autour d'un niveau qu'autour d'un inter-niveau car sa plage de variation est dans tous les cas inférieure à celle d'un inter-niveau.

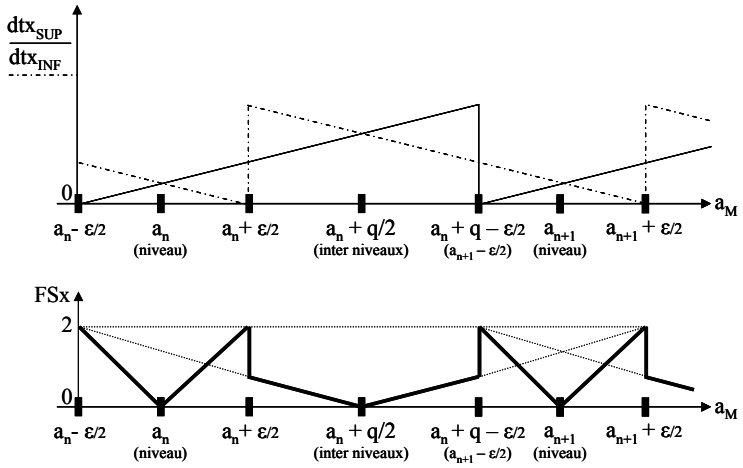


Figure 20 : Evolutions schématiques des intervalles de temps des crêtes et du facteur de symétrie en fonction de la valeur moyenne

Pour illustrer toutes les considérations théoriques qui ont été abordées dans les sections précédentes, nous proposons de simuler l'échantillonnage par traversée de niveaux d'un signal sinusoïdal de valeur moyenne a_M variable et d'amplitude crête constante $a_o = 0.5V$. La plage de variation de a_M est fixée de telle manière que le signal analogique ne soit jamais saturé c'est-à-dire

qu'il ne sorte jamais de la dynamique du convertisseur. Nous proposons donc d'étudier la valeur moyenne dans un intervalle $[a_o; V_{alim}-a_o]$. Par ailleurs, comme la théorie ne tenait pas compte de la quantification du temps, introduite par le timer (ce qui permettait d'avoir des intervalles de temps réels positifs ou nuls), nous utiliserons un modèle de convertisseur idéal (le pas temporel du timer est nul, les intervalles de temps sont exacts) 4 bits utilisant 15 niveaux régulièrement espacés sur la dynamique.

La Figure 21 représente l'évolution des intervalles de temps des crêtes en fonction de la valeur de moyenne du signal. Comme nous l'avons vu dans les sections précédentes, ils ont des variations opposées en en arc sinus et se croisent à tous les niveaux et inter-niveaux.

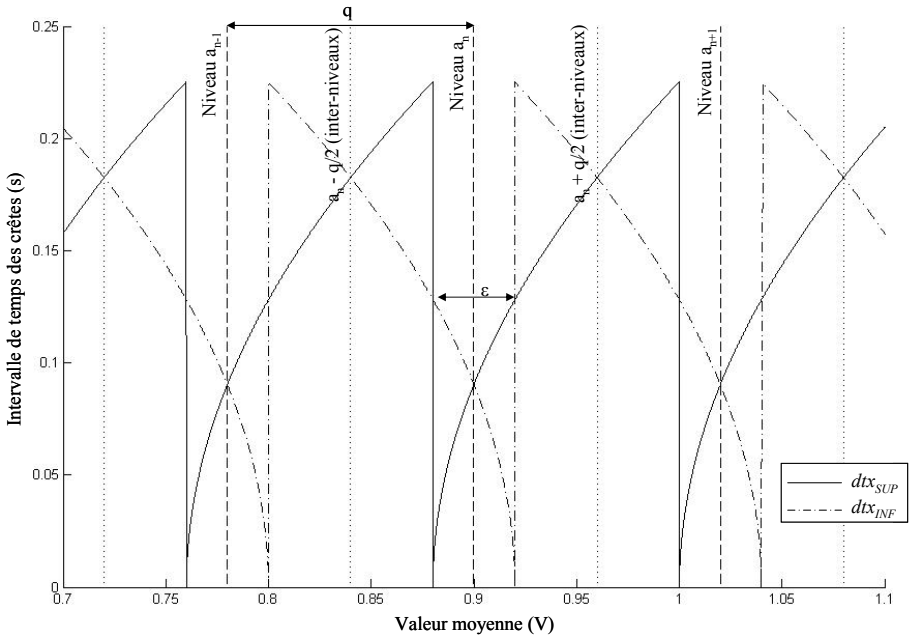


Figure 21 : Exemple d'évolution simulée des intervalles de temps en fonction de la valeur moyenne

La Figure 22 montre l'évolution du facteur de symétrie en fonction de la valeur moyenne du signal analogique. Nous retrouvons l'allure étudiée dans les sections précédentes et illustrée sur la Figure 20 : autour d'un niveau, le facteur de symétrie évolue non linéairement selon une loi approchant un en arc sinus tandis qu'il augmente linéairement autour d'un inter-niveau.

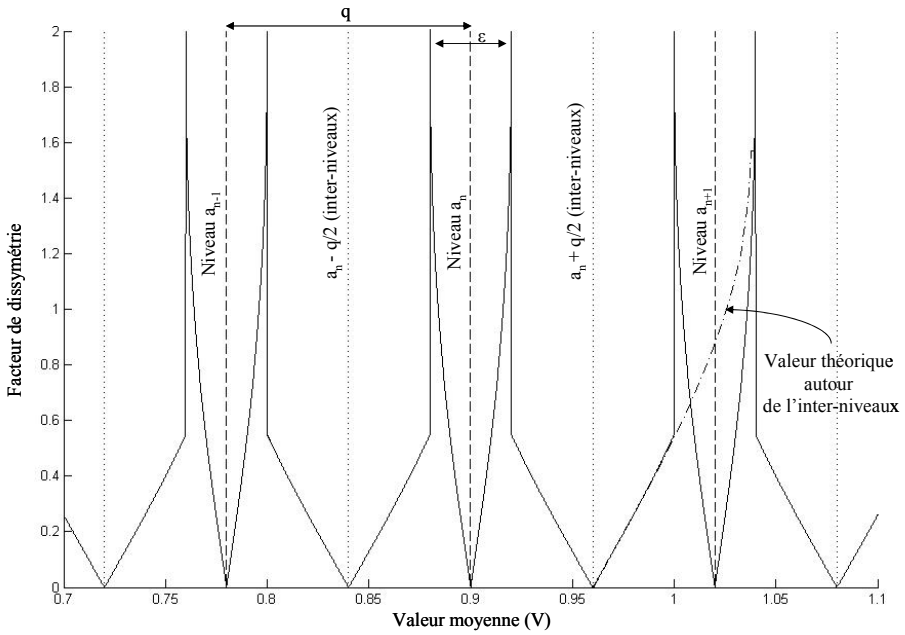


Figure 22 : Exemple d'évolution simulée du facteur de symétrie en fonction de la valeur moyenne

II.1.2.4 Conclusion : impact de la symétrie sur l'échantillonnage

D'après tout ce que nous venons d'étudier, nous pouvons dégager deux situations :

- L'échantillonnage est dissymétrique
- L'échantillonnage est symétrique

Au vu de l'évolution du facteur de symétrie, la première situation est majoritaire étant donnée que le facteur est non nul dans tous les cas excepté en un nombre fini de positions i.e. lorsque la valeur moyenne est un niveau ou un inter-niveau. Dans ce cas, l'ensemble des intervalles de temps de la partie supérieure de la sinusoïde est différent de celui de la partie inférieure. Ainsi les intervalles de temps et donc également les instants d'échantillonnage, sont périodiques de période T_o tels qu'ils ont été étudiés dans le cas général.

En revanche, la seconde situation implique que l'ensemble des intervalles de temps de la partie supérieure de la sinusoïde est égal à celui de la partie inférieure. Ainsi, les intervalles de temps et donc également les instants d'échantillonnage sont périodiques de période $T_o/2$. Ceci aura des répercussions sur le schéma d'échantillonnage et le spectre du signal échantillonné que nous étudierons dans la section suivante.

II.1.3 Exemples de représentations d'un signal sinusoïdal

Nous proposons dans cette section de présenter deux exemples d'échantillonnage par traversée de niveaux de signaux sinusoïdaux. Les deux signaux sont de fréquence $f_o = 1Hz$, ont la même amplitude crête mais ont deux valeurs moyennes différentes. Dans le premier cas, la valeur moyenne rend l'échantillonnage dissymétrique tandis que dans le second, la valeur moyenne centre le signal sur la grille de niveaux. Les Figure 23 et Figure 24 représentent respectivement l'amplitude et les intervalles de temps du signal échantillonné.

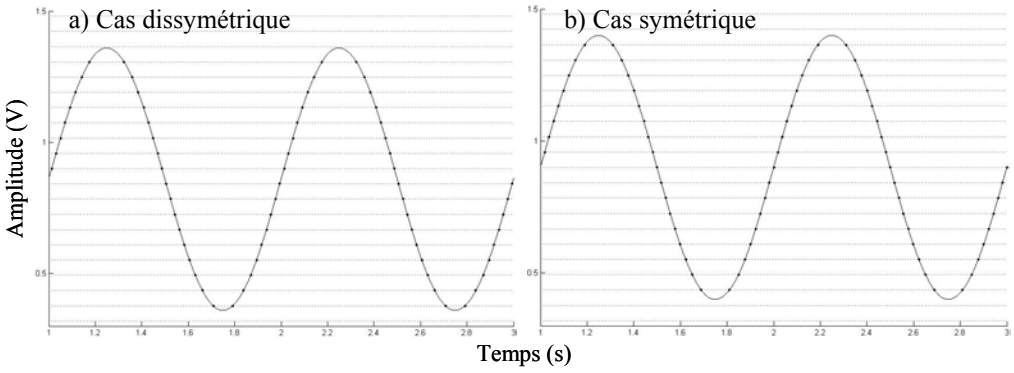


Figure 23 : Amplitude d'un signal sinusoïdal échantillonné par traversée de niveaux.

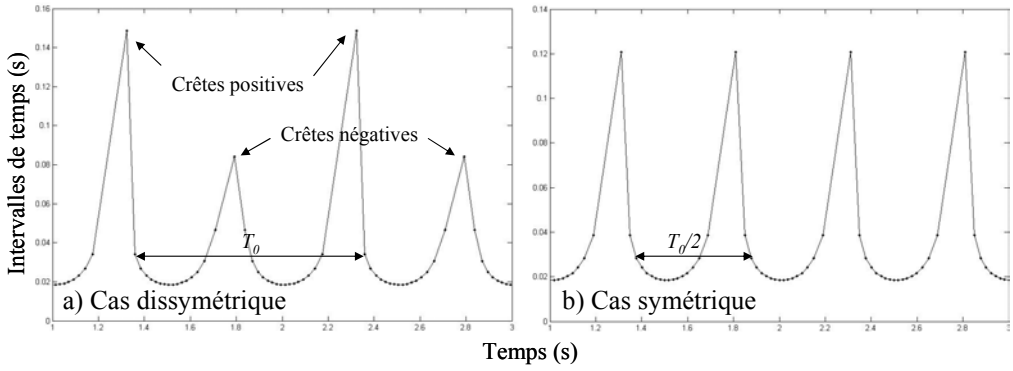


Figure 24 : Intervalles de temps d'un signal sinusoïdal échantillonné par traversée de niveaux.

II.1.4 Schéma d'échantillonnage et spectre d'un signal sinusoïdal

Pour un signal sinusoïdal de fréquence f_o, nous avons montré que les intervalles de temps et les instants d'échantillonnage sont des signaux périodiques de même période que le signal d'entrée. La fonction d'échantillonnage FE_x, i.e. un peigne de Dirac non uniforme dont les instants dépendent du signal, est donc aussi périodique. Le schéma d'échantillonnage SE_x se calcule en décomposant FE_x en série de Fourier :

$$\begin{aligned} FE_x(t) &= \sum_{n\in\mathbf{Z}} \delta(t - tx_n) \\ &= \sum_{n\in\mathbf{Z}} c_n e^{-j2\pi n f_o t} \end{aligned} \qquad \text{Eq. (52)}$$

$$\begin{aligned} SE_x(f) &= TF[FE_x] \\ &= \sum_{n\in\mathbf{Z}} c_n \delta(f - nf_o) \end{aligned} \qquad \text{Eq. (53)}$$

SE_x est donc un spectre composé de raies de fréquence fondamentale f_o dont les coefficients dépendent du signal d'entrée. Si l'échantillonnage est symétrique, la période de la fonction d'échantillonnage est $T_o/2$; les coefficients c_{2n+1} sont nuls pour tout $n \in \mathbf{Z}$. Par ailleurs, comme les intervalles de temps sont quantifiés avec une résolution T_c, la fonction d'échantillonnage n'est pas périodique de période T_o mais $T_o \pm T_C$. Pour simplifier l'étude, nous supposons que la résolution du timer est un sous-multiple de la fréquence fondamentale – l'erreur est alors compensée par une petite variation de la valeur des coefficients c_n. Dans ce cas, le schéma d'échantillonnage est périodique de période $F_c = 1/T_c$ [Bagshaw *et al.* 1991]. Enfin, il est important de noter que les coefficients c_n ne dépendent pas de la fréquence du signal f_o. En effet, si les amplitudes crêtes et moyennes sont fixées, pour un même convertisseur, nous avons vu que les intervalles de temps sont proportionnels à la période T_o du signal. La répartition des impulsions de Dirac au sein d'une période est donc la même quelle que soit la valeur de T_o. Comme les coefficients c_n caractérisent une série de Fourier de fréquence fondamentale f_o, la valeur des coefficients est donc indépendante de la valeur de f_o.

En reprenant les deux exemples précédents, la Figure 25 illustre le schéma d'échantillonnage d'un signal sinusoïdal dans les deux configurations possibles. Sur la partie gauche, l'échantillonnage n'étant pas symétrique, tous les coefficients sont non nuls. En revanche,

sur la partie droite, la périodicité des intervalles de temps est réduite à $T_o/2$; les coefficients impairs se sont annulés.

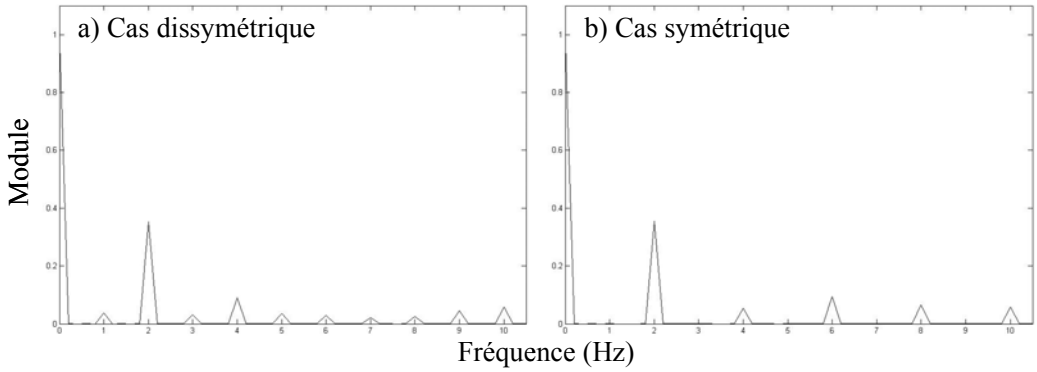


Figure 25 : Schémas d'échantillonnage de signaux sinusoïdaux

En élargissant le plan fréquentiel, on peut également observer la périodicité du schéma d'échantillonnage. Sur l'exemple présenté sur la Figure 26, le timer avait une résolution $F_C = 100Hz$.

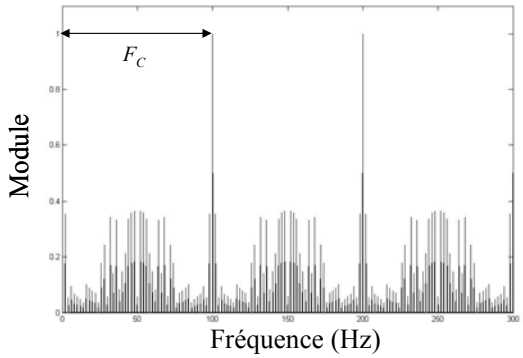


Figure 26 : Plan large du schéma d'échantillonnage d'un signal sinusoïdal

Le spectre du signal sinusoïdal échantillonné par traversée de niveaux peut donc être calculé par le produit de convolution entre le spectre du signal analogique et le schéma d'échantillonnage. Comme ces derniers sont tous deux des trains d'impulsions de même fréquence fondamentale f_o et comme l'impulsion de Dirac est l'élément neutre du produit de convolution, le spectre du signal échantillonné est l'accumulation du spectre du signal analogique décalé à toutes les harmoniques de f_o et pondéré par l'amplitude du schéma d'échantillonnage de chaque

harmonique. Mathématiquement, il s'exprime par les relations de l'équation (54) qui sont illustrées sur la Figure 28.

$$\begin{aligned} X_e(f) &= (X * SE_x)(f) \\ &= \left(a_M + j\frac{a_o}{2}\left(\delta(f+f_o) - \delta(f-f_o)\right)\right) * SE_x(f) \\ &= a_M SE_x(f) + j\frac{a_o}{2}\left(SE_x(f+f_o) - SE_x(f-f_o)\right) \\ &= \sum_{n\in\mathbf{Z}} \left(a_M c_n + j\frac{a_o}{2}\left(c_{n+1} - c_{n-1}\right)\right)\delta(f-nf_o) \\ &= \sum_{n\in\mathbf{Z}} \chi_n \delta(f-nf_o) \end{aligned} \qquad \text{Eq. (54)}$$

Le spectre du signal échantillonné est donc de la même forme que le schéma d'échantillonnage : il s'agit d'un spectre de raies de fréquence fondamentale f_o dont les amplitudes dépendent à la fois de celles du signal et des instants d'échantillonnage, c'est-à-dire des variations du signal. Le spectre du signal échantillonné est par ailleurs périodique de période F_c à cause de la quantification du temps [Aeschlimann *et al.* 2005] [Aeschlimann *et al.* 2005].

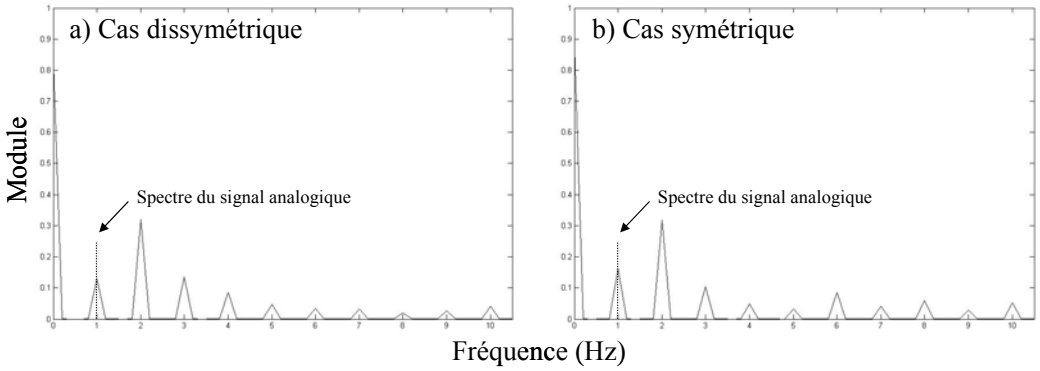


Figure 27 : Spectre de signaux sinusoïdaux échantillonnés par traversée de niveaux

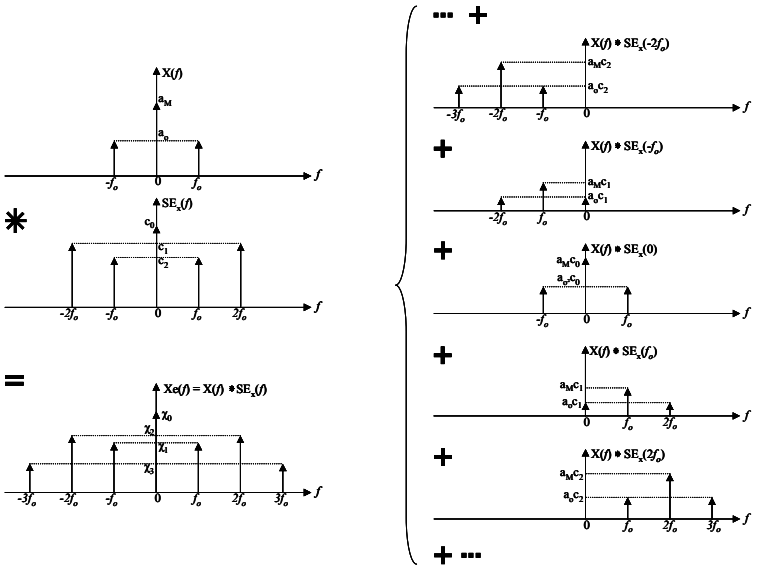


Figure 28 : Produit de convolution entre le spectre d'un signal sinusoïdal à temps continu et son schéma d'échantillonnage par traversée de niveaux

En conclusion, l'échantillonnage par traversée de niveaux d'un signal sinusoïdal induit obligatoirement un repliement de spectre. Il n'est donc pas possible de retrouver les paramètres du signal analogique (amplitudes a_M et a_o et fréquence fondamentale f_o) à partir du signal échantillonné en étudiant sa transformée de Fourier comme le montre la Figure 27.

II.2 Généralisation : échantillonnage d'un signal périodique quelconque

Cette partie a pour objet la généralisation du cas de l'échantillonnage d'un signal sinusoïdal au cas de tous les signaux périodiques de période fondamentale $T_o = 1/f_o$, décomposable par série de Fourier en somme de sinusoïdes :

$$x(t) = \sum_{n \in \mathbf{Z}} a_n e^{-j\pi n f_o t} \qquad \text{Eq. (55)}$$

II.2.1 Etude des intervalles de temps

A la différence du signal sinusoïdal, il est impossible de déterminer les intervalles de temps d'un signal périodique en fonction des paramètres du signal pour un convertisseur donné car la fonction n'est pas bijective sur un intervalle de temps unique.

En revanche, comme pour les signaux sinusoïdaux, étant donné que la dérivée du signal est proportionnelle à la fréquence fondamentale, les intervalles de temps sont aussi proportionnels à la période du signal :

$$\frac{dx(t)}{dt} = -j2\pi f_o \sum_{n\in\mathbf{Z}} n \cdot a_n e^{-j2\pi n f_o t} \qquad \text{Eq. (56)}$$

De plus, l'équation (56) montre que la dérivée d'un signal périodique est également périodique. Les intervalles de temps, image des variations du signal, sont donc également périodiques de période T_o.

Enfin, du fait de la quantification du temps, les intervalles de temps sont tous multiples de T_c, la résolution du timer. Nous supposons que T_C est à nouveau un sous-multiple de T_o

II.2.2 Symétrie

Dans le cas d'un signal périodique, l'amplitude peut varier symétriquement de part et d'autre de la valeur moyenne en fonction des composantes présentes dans le signal. Pour cela, il suffit que la valeur absolue du signal ôté de sa valeur moyenne soit périodique de période $T_o/2$. En séparant les composantes paires et impaires du signal, la condition devient :

$$\begin{aligned}\left|x\left(t+\frac{T_o}{2}\right)-a_0\right| &= \left|\sum_{n\in\mathbf{Z}^*} a_{2n} e^{-j2\pi(2n)f_o t} e^{-j2\pi(2n)f_o\frac{T_o}{2}} + \sum_{n\in\mathbf{Z}} a_{2n+1} e^{-j2\pi(2n+1)f_o t} e^{-j2\pi(2n+1)f_o\frac{T_o}{2}}\right| \\ &= \left|\sum_{n\in\mathbf{Z}^*} a_{2n} e^{-j2\pi(2n)f_o t} - \sum_{n\in\mathbf{Z}} a_{2n+1} e^{-j2\pi(2n+1)f_o t}\right| \qquad \text{Eq. (57)} \\ &= \left|\sum_{n\in\mathbf{Z}^*} a_{2n} e^{-j2\pi(2n)f_o t} + \sum_{n\in\mathbf{Z}} a_{2n+1} e^{-j2\pi(2n+1)f_o t}\right|\end{aligned}$$

La symétrie est vérifiée si et seulement si l'un des deux membres est nul : si $a_{2n} = 0$ pour tout $n \in \mathbf{Z}^*$ ou si $a_{2n+1} = 0$ pour tout $n \in \mathbf{Z}$. Or par définition, la composante fondamentale est toujours

présente c'est-à-dire $a_1 \neq 0$. Pour que le signal soit symétrique il faut donc que le signal n'ait que des composantes impaires. Il existe alors trois cas possibles pour conclure sur la symétrie de l'échantillonnage par traversée de niveaux d'un signal périodique :

- Le signal n'a que des composantes impaires et sa valeur moyenne est un niveau ou inter-niveau : les intervalles de temps sont périodiques de période $T_o/2$.
- Le signal n'a que des composantes impaires et sa valeur moyenne n'est pas un niveau ou inter-niveau : les intervalles de temps sont périodiques de période T_o.
- Le signal possède des composantes paires : les intervalles de temps sont périodiques de période T_o.

II.2.3 Exemples de représentations d'un signal périodique

Nous proposons d'illustrer les trois types de signaux périodiques conduisant au trois cas de symétrie lors de l'échantillonnage par traversée de niveaux. Dans les deux premiers exemples présentés sur les Figure 29 et Figure 30, il s'agit d'un signal à deux composantes impaires : la fréquence fondamentale ($f_o = 1Hz$) et son harmonique 3. La valeur moyenne du signal change entre les deux cas pour rendre respectivement l'échantillonnage symétrique puis dissymétrique.

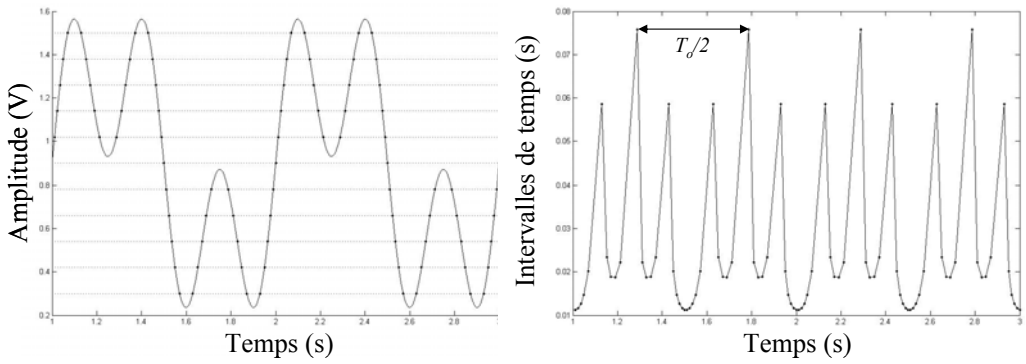


Figure 29 : Exemple de signal périodique symétrique échantillonné symétriquement

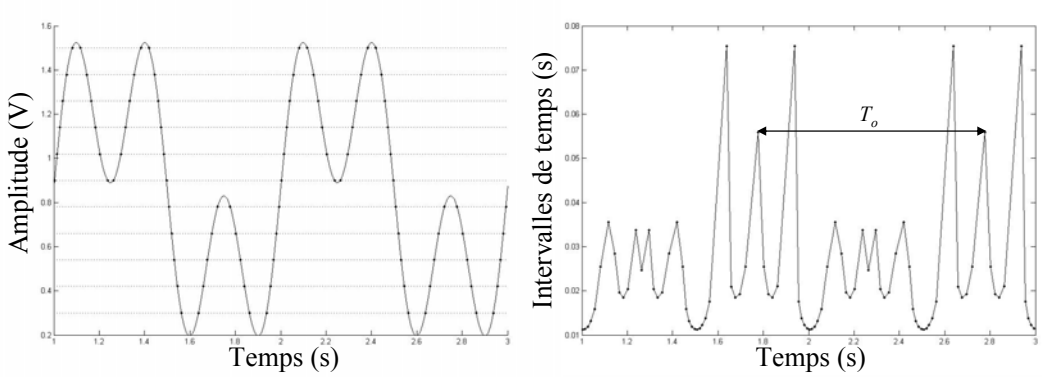


Figure 30 : Exemple d'un signal périodique symétrique échantillonné non symétriquement

Dans le troisième exemple présenté sur la Figure 31, il s'agit d'un signal composé de la somme des six premières harmoniques dont la fréquence fondamentale est $f_o = 1Hz$. La présence de composantes paires rend le signal dissymétrique ; l'échantillonnage ne pouvant pas être symétrique, les intervalles de temps sont périodiques de période T_o.

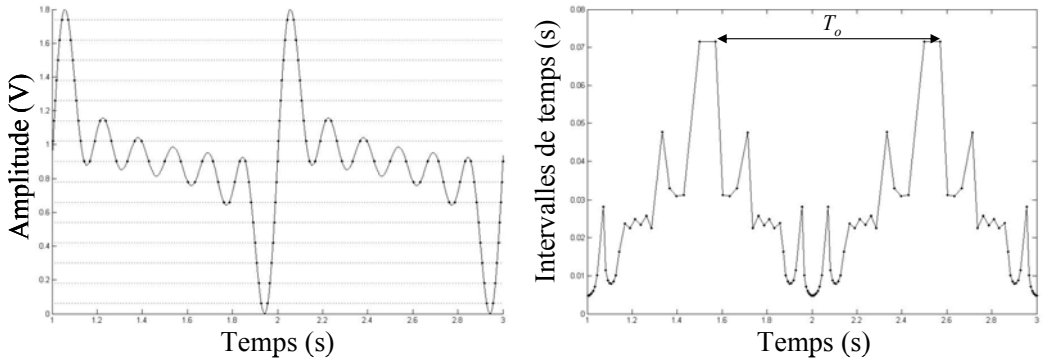


Figure 31 : Exemple d'un signal périodique non symétrique. L'échantillonnage est également non symétrique.

II.2.4 Schéma d'échantillonnage et spectre d'un signal périodique

Nous savons que si le signal est de période fondamentale $T_o = 1/f_o$, les intervalles de temps seront au moins périodiques de période T_o. La fonction d'échantillonnage FE_x est donc à nouveau un peigne de Dirac non uniforme périodique de période T_o (équation (52)), et le schéma

d'échantillonnage, un spectre de raies de fréquence fondamentale f_o et de période F_c dont les amplitudes dépendent des variations du signal (équation (53)). Le spectre du signal échantillonné, obtenu par produit de convolution, est par conséquent la somme du spectre du signal analogique décalé à toutes les harmoniques de f_o et pondéré par l'amplitude du schéma d'échantillonnage de chaque harmonique (Figure 32) :

$$\begin{aligned} X_e(f) &= (X * SE_x)(f) \\ &= \sum_{m\in\mathbf{Z}} c_m \delta(f - nf_o) * X(f) \\ &= \sum_{m\in\mathbf{Z}} c_m X(f - nf_o) \\ &= \sum_{m\in\mathbf{Z}} c_m \sum_{n\in\mathbf{Z}} a_n \delta((f - mf_0) - nf_0) \\ &= \sum_{m\in\mathbf{Z}} \sum_{n\in\mathbf{Z}} a_n c_m \delta(f - (n+m) f_0) \\ &= \sum_{m\in\mathbf{Z}} \chi_k \delta(f - kf_0) \end{aligned} \qquad \text{Eq. (58)}$$

L'amplitude de chaque harmonique χ_k est une combinaison des harmoniques du signal (équation (59)). Le spectre du signal échantillonné est donc à nouveau de la même forme que celle du spectre du signal analogique et celle du schéma d'échantillonnage. En revanche, il est encore replié du fait de la combinaison linéaire [Aeschlimann *et al.* 2005], [Aeschlimann *et al.* 2005].

$$\chi_k = \sum_{n\in\mathbf{Z}} \sum_{m=k-n}^{+\infty} a_n c_m \qquad \text{Eq. (59)}$$

Les équations (54) et (58) étant les mêmes, le schéma d'échantillonnage et le spectre d'un signal périodique échantillonné par traversée de niveaux peuvent être illustrés par les Figure 25, Figure 26 et Figure 27 obtenues dans le cas d'un signal sinusoïdal.

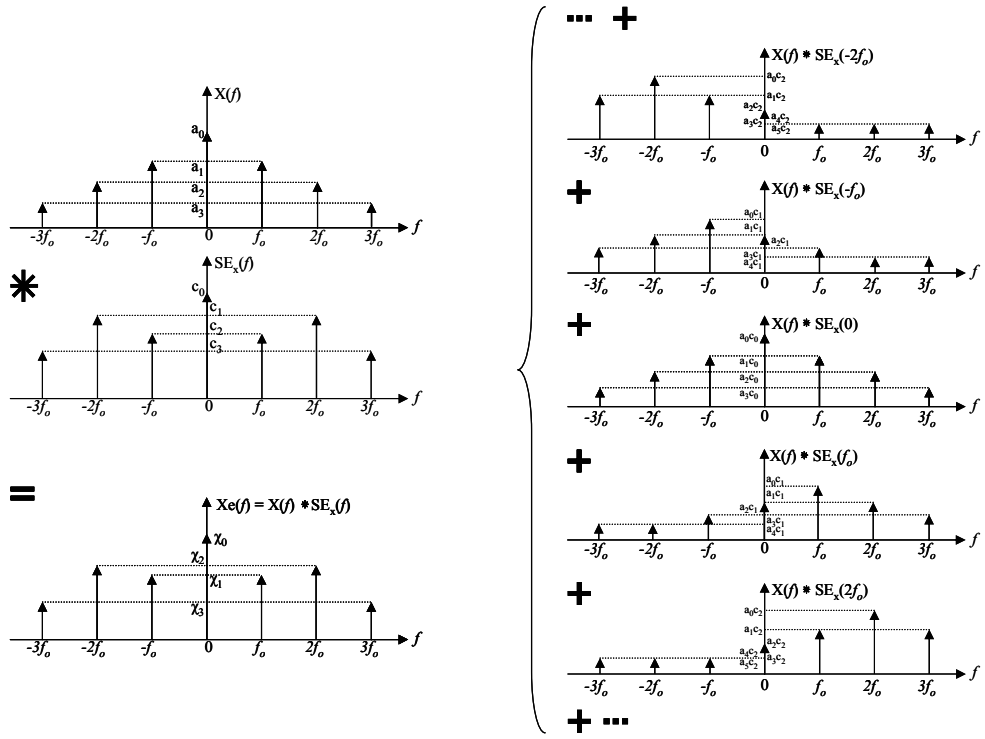


Figure 32 : Spectre d'un signal périodique échantillonné par traversée de niveaux

En conclusion, l'échantillonnage par traversée de niveaux d'un signal périodique entraîne obligatoirement un repliement de spectre. Il n'est donc pas possible d'analyser un signal analogique à partir de la transformée de Fourier du signal échantillonné.

CHAPITRE III

Etude de l'échantillonnage par traversée de niveaux des signaux non périodiques

---

Dans le chapitre précédent, nous avions abordé l'échantillonnage par traversée de niveaux des signaux périodiques. Dans la mesure où leurs variations sont également périodiques et que l'échantillonnage par traversée de niveaux s'adapte aux variations du signal, les propriétés de l'échantillonnage (la répartition des échantillons dans le temps, la fréquence moyenne d'échantillonnage…) sont constantes dans le temps. Nous décidons d'étudier dans ce chapitre des signaux usuels dont les propriétés varient au cours du temps pour analyser les mécanismes d'adaptation de l'échantillonnage.

Dans un premier temps, nous considérons un signal impulsionnel gaussien et un signal composé d'une succession d'impulsions. Nous montrons que les propriétés du schéma d'échantillonnage évoluent au cours du temps : il bascule en effet alternativement entre un peigne de Dirac régulier pendant les zones inactives et un peigne de Dirac non uniforme qui dépend de la forme de l'impulsion. Dans un second temps, nous considérons un signal non stationnaire. Pour se ramener à un cas connu, nous utilisons une modulation linéaire de fréquence. Nous généralisons le cas précédent en montrant que le schéma d'échantillonnage s'adapte automatiquement au contenu spectral de la zone en cours de conversion. Dans tous les cas, le schéma d'échantillonnage conduit à un repliement du spectre du signal échantillonné, ce qui nous amène à conclure que le traitement d'un signal échantillonné non uniformément par traversée de niveaux doit utiliser obligatoirement l'information portée par les intervalles pour pondérer la valeur des amplitudes.

III.1 Echantillonnage d'un signal impulsionnel

III.1.1 Choix du signal

Un signal impulsionnel quel qu'il soit, possède un spectre continu en fréquence. Nous pouvons alors distinguer deux catégories de signaux : ceux à durée finie dont le spectre est à bande illimitée et ceux à durée infinie dont le spectre est à bande limitée. Dans tous les cas un signal, doit respecter une relation d'incertitude par analogie avec le principe d'incertitude d'Heisenberg rencontré en mécanique quantique :

$$B_u T_u \geq \frac{4}{\pi} \qquad \text{Eq. (60)}$$

où B_u représente la bande utile du spectre et T_u la durée utile du signal.

Il existe un signal pour lequel cette relation est atteinte : appelé *logon* par Gabor, il s'agit d'une impulsion en cloche de Gauss à durée et bande utiles limitées :

$$\begin{cases} x(t) = a_0 e^{-\alpha t^2} \\ X(f) = a_0 \sqrt{\dfrac{\pi}{\alpha}} e^{-\frac{\pi^2 f^2}{\alpha}} \end{cases} \qquad \text{Eq. (61)}$$

La bande passante B_u et la durée T_u du signal sont alors définies par :

$$\begin{cases} B_u = \dfrac{\sqrt{2\alpha}}{\pi} \\ T_u = \dfrac{2\sqrt{2}}{\sqrt{\alpha}} \end{cases} \qquad \text{Eq. (62)}$$

Cette fonction est notamment utilisée en communication numérique pour concevoir certaines modulations à spectre très étroit dans le cadre de transmissions multiplexées en fréquence. En effet, comme elle est indéfiniment dérivable, son spectre possède la propriété de décroître plus rapidement que toute puissance de f [Delmas 1991].

III.1.2 Etude des intervalles de temps

Pour un niveau donné a_n parmi les N niveaux possibles, l'amplitude du signal permet de calculer les instants d'échantillonnage tx_i :

$$tx_i^{\,2} = \frac{1}{\alpha}\ln\left(\frac{a_0}{a_n}\right) \qquad \text{Eq. (63)}$$

L'équation (63) montre que pour un niveau, il y a seulement deux instants d'échantillonnage opposés (du fait de la parité de la fonction) et donc deux points prélevés : l'un lors de la phase croissante du signal et l'autre lors de la phase décroissante :

$$\begin{cases} tx_i = \sqrt{\frac{1}{\alpha}\ln\left(\frac{a_0}{a_n}\right)} \\ tx_j = tx_{-i} = -\sqrt{\frac{1}{\alpha}\ln\left(\frac{a_0}{a_n}\right)} \end{cases} \qquad \text{Eq. (64)}$$

Lors des deux transitions entre deux niveaux successifs a_n et a_{n-1} (passage de a_{n-1} vers a_n ou bien a_n vers a_{n-1}), il existe donc deux intervalles de temps de même valeur notée dtx_n :

$$dtx_n = \sqrt{\frac{1}{\alpha}\ln\left(\frac{a_0}{a_n}\right)} - \sqrt{\frac{1}{\alpha}\ln\left(\frac{a_0}{a_n - q}\right)} \qquad \text{Eq. (65)}$$

Toutefois, nous pouvons observer une exception concernant le premier niveau. En effet, le premier point échantillonné a un intervalle de temps infini. Ceci implique que ce point n'existe pas puisqu'un échantillon est défini par un couple amplitude, intervalle de temps. Cet état est facilement remarquable sur la Figure 33.

En revanche, lors du passage par la crête du signal, le signal croise deux fois le même niveau a_N , échantillonnant les $N^{ième}$ et $(N+1)^{ième}$ points pendant une durée unique dtx_{N+1} :

$$dtx_{N+1} = 2\sqrt{\frac{1}{\alpha}\ln\left(\frac{a_0}{a_N}\right)} \qquad \text{Eq. (66)}$$

Finalement, ces résultats montrent que l'échantillonnage par traversée de niveaux d'un signal impulsionnel induit la création d'un nombre fini de points bien que la fonction soit définie

pour tout $t \in \mathbf{R}$. Cette remarque est d'autant plus importante qu'il s'agit là de la seule technique obtenant ce résultat.

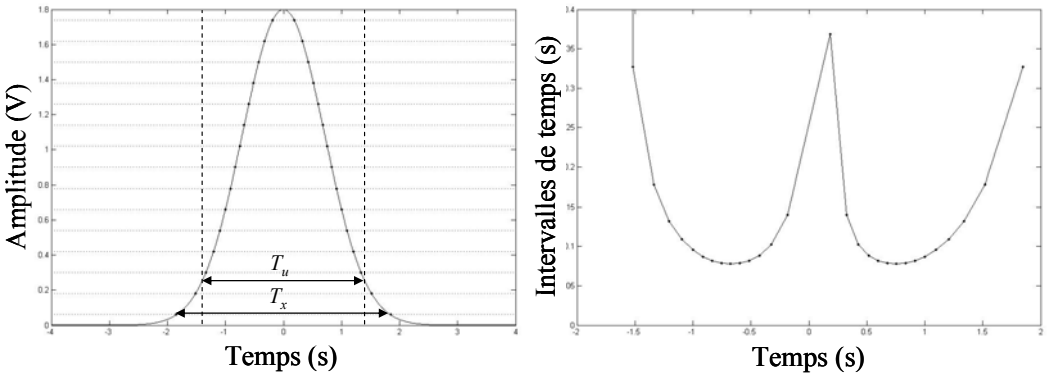


Figure 33 : Echantillonnage par traversée de niveaux d'un signal impulsionnel

III.1.3 Schéma d'échantillonnage et spectre d'un signal impulsionnel

III.1.3.1 Schéma d'échantillonnage

Nous savons que pour un signal impulsionnel, le signal échantillonné est composé de *2N* points où *N* correspond au nombre de niveaux traversés. La fonction d'échantillonnage est donc un peigne de Dirac non uniforme à durée limitée :

$$FE_x(t) = \sum_{n=1}^{N} \left(\delta(t + tx_n) + \delta(t - tx_n) \right) \qquad \text{Eq. (67)}$$

En calculant la transformée de Fourier de FE_x, nous pouvons en déduire que le schéma d'échantillonnage est une fonction continue en fréquence et oscillante :

$$\begin{aligned} SE_x(f) &= TF\left[FE_x\right] \\ &= \sum_{n=1}^{N} \left(e^{j2\pi f t x_n} + e^{-j2\pi f t x_n} \right) \\ &= 2\sum_{n=1}^{N} \cos(2\pi f t x_n) \end{aligned} \qquad \text{Eq. (68)}$$

Toutefois, comme la durée du peigne est limitée, la fonction d'échantillonnage ne peut plus être considérée comme l'échantillonnage de la fonction unité mais plutôt comme l'échantillonnage

d'une fonction porte dont la largeur $T_x = 2tx_1$ dépend du signal. Notons que T_x est de l'ordre de grandeur du support temporel T_u du signal analogique et généralement supérieur $T_x > T_u$. Par ailleurs, il est possible d'interpréter également la durée limitée comme la troncature d'une fonction d'échantillonnage illimitée *FE* sur une durée T_x. Dans ce cas, le schéma d'échantillonnage est obtenu par le produit de convolution entre la transformée de Fourier de la fonction d'échantillonnage illimitée et la transformée de Fourier de la fenêtre. Bien entendu, la fonction illimitée nous est inconnue puisqu'elle n'existe pas. Cependant, si nous supposons que son support temporel est infini ou du moins qu'il est beaucoup plus grand que T_x, nous pouvons approcher son spectre par une fonction de support illimité mais dont la bande passante est beaucoup plus étroite que le lobe principal du sinus cardinal. Ainsi dans une première approximation, le schéma d'échantillonnage pourrait être assimilé à un sinus cardinal dont l'évolution est déformée :

$$\begin{cases} FE_x(f) = FE(t) \cdot \prod_{T_x}(t) \\ SE_x(f) = (SE * \mathrm{sinc}(\pi T_x) * (f) \end{cases} \qquad \text{Eq. (69)}$$

où $\prod_{T_x}(t)$ est la fonction porte de largeur T_x. Enfin, comme les intervalles de temps sont quantifiés avec une résolution T_C, le schéma d'échantillonnage est périodique de période $F_C = 1/T_C$.

III.1.3.2 Spectre d'un signal impulsionnel échantillonné

Le spectre du signal échantillonné est obtenu par le produit de convolution entre le spectre du signal analogique et le schéma d'échantillonnage. D'après l'équation (68), il peut s'écrire :

$$\begin{aligned} X_e(f) &= X(f) * SE_x(f) \\ &= 2X(f) * \sum_{n=1}^{N} \cos(2\pi f t x_n) \end{aligned} \qquad \text{Eq. (70)}$$

A partir de l'équation (69), il est également possible d'interpréter le spectre du signal échantillonné de la manière suivante :

$$X_e(f) = X(f) * SE(f) * \mathrm{sinc}(\pi f T_x) \qquad \text{Eq. (71)}$$

Comme il est difficile de calculer formellement le spectre du signal échantillonné, nous avons à nouveau vérifié nos hypothèses à l'aide d'expériences sous *Matlab*. Ainsi sur la Figure 33, nous présentons une simulation dans laquelle une impulsion gaussienne est échantillonnée par

traversée de niveaux ; les paramètres du signal sont $a_0 = 1.8V$ et $\alpha = 1$. La durée utile du signal vaut alors $T_u = 2.83s$ tandis que la bande passante est égale à $B_u = 0.45Hz$. L'échantillonnage est effectué sur 15 niveaux répartis uniformément dans la dynamique $[0 ; a_0]$. Les intervalles de temps sont mesurés par un timer de résolution $F_c = 20Hz$.

Dans le domaine fréquentiel, nous représentons sur la Figure 34 le module du schéma d'échantillonnage normalisé ; sur la partie gauche (zoom dans la bande [0 ; 1Hz]), on peut remarquer qu'il s'agit bien d'une fonction continue en fréquence ressemblant à un sinus cardinal dont la largeur du lobe principal est $2/T_x$ (trait hachuré). En outre, le premier creux du schéma d'échantillonnage est légèrement supérieur à $1/T_x$. Sur la partie droite, nous pouvons noter que le schéma d'échantillonnage est bien une fonction continue en fréquence, oscillante et périodique de période F_c. Sur la Figure 35 sont représentés les spectres des signaux analogique (hachuré) et échantillonné (trait plein). La convolution introduite par l'opération d'échantillonnage déforme le spectre initial. En effet, étant donné que la largeur du spectre B_u est plus grande que le lobe principal du schéma d'échantillonnage, la convolution *privilégie* le spectre dont la bande est la plus large i.e. celui du signal analogique mais pas au point de conserver intégralement la forme du spectre dans la bande passante. Il laisse apparaître en effet le creux du lobe principal du schéma d'échantillonnage cardinal décalé vers les hautes fréquences de la bande passante.

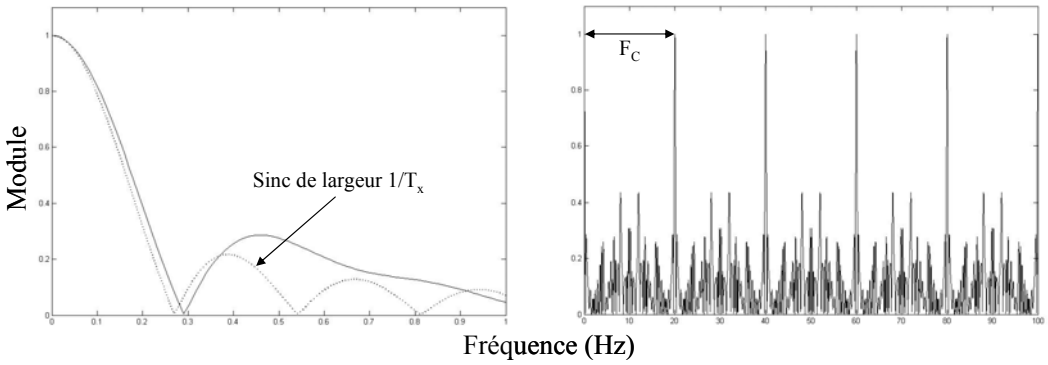


Figure 34 : Schéma d'échantillonnage normalisé d'un signal impulsionnel

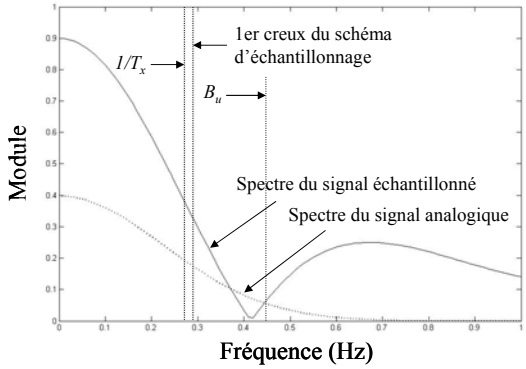


Figure 35 : Spectre d'un signal impulsionnel échantillonné par traversée de niveaux

Pour résoudre ce problème, nous savons intuitivement qu'en augmentant la résolution du convertisseur (donc en diminuant la valeur du quantum), la fenêtre d'observation devient plus large réduisant par la même occasion la largeur du lobe principal. Le lobe principal du schéma d'échantillonnage serait alors réduit et conduirait le produit de convolution à tendre vers le spectre du signal analogique. Pour déterminer la valeur de la résolution du convertisseur à appliquer, nous exprimons le rapport de la largeur du lobe principal sur la bande passante du signal. En considérant un échantillonnage sur *N* niveaux et une impulsion d'amplitude égale à la pleine échelle, la largeur du lobe principal est définie d'après l'équation (64) par :

$$\frac{1}{T_x} = \frac{1}{2tx_1} = \frac{1}{2}\sqrt{\frac{\alpha}{\ln\left(\frac{a_0}{a_1}\right)}} = \frac{1}{2}\sqrt{\frac{\alpha}{\ln(2N)}} \qquad \text{Eq. (72)}$$

car $a_1 = q/2 = (a_0/N)/2$. Le rapport entre de la largeur du lobe principal sur la bande passante du signal peut alors s'écrire :

$$\frac{1}{B_u T_x} = \frac{\pi}{2\sqrt{2}}\frac{1}{\sqrt{\ln(2N)}} \qquad \text{Eq. (73)}$$

Nous pouvons remarquer que le rapport ne dépend que du nombre de niveaux *N*, donc de la résolution du convertisseur. Sur la Figure 36, nous avons évalué l'équation (73) pour différentes résolutions du convertisseur. Pour des valeurs raisonnables allant de 2 bits à 6 bits, le rapport est

toujours supérieur à 0,5. Pour obtenir un lobe principal dix fois plus petit que la bande passante du signal, il faut augmenter la résolution à une valeur irréaliste de 174 bits. Ceci montre donc que pour une résolution raisonnable, le schéma d'échantillonnage induira systématiquement un repliement du spectre du signal analogique dans sa bande passante.

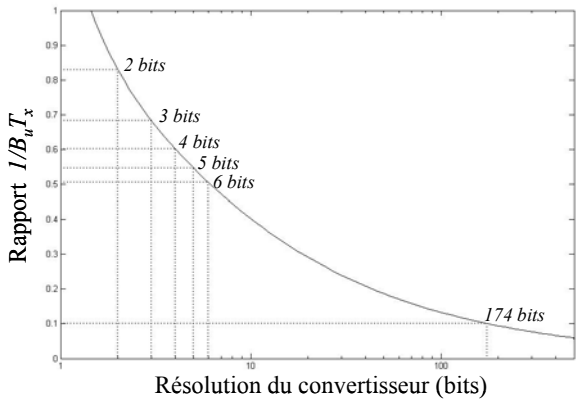


Figure 36 : Rapport de la largeur du lobe principal du schéma d'échantillonnage sur la bande passante du signal en fonction de la résolution du convertisseur

Par ailleurs, si nous observons les spectres en $f = 0Hz$, nous pouvons noter une grande différence dans l'amplitude de la composante continue. Pour le signal analogique, ce point correspond à sa valeur moyenne tandis que pour le signal échantillonné, il correspond à la valeur moyenne des amplitudes prélevées. Or comme il y a $2N$ points répartis sur les N niveaux entre $0V$ et a_0, cette valeur est égale à $a_0/2$ soit $0.9V$ dans notre exemple.

D'une manière générale, la déformation vient du fait que le spectre, c'est-à-dire la transformée de Fourier du signal échantillonné, est calculé à partir des amplitudes des points. Ainsi, dans le calcul, la même valeur informative est attribuée à chaque échantillon et ce quelle que soit la distance qui le sépare de ses voisins. Or évidemment, pour obtenir un résultat cohérent, il faut pondérer l'information des amplitudes par l'information portée par les intervalles de temps. Nous tiendrons compte de ce constat dans les chapitres suivants dédiés à la reconstruction et aux outils de traitements.

III.1.3.3 Schéma d'échantillonnage d'un signal impulsionnel avec saturation du timer

Dans la section précédente, nous faisions l'hypothèse que le premier point n'était obtenu qu'à partir d'un instant $t = -tx_1$, impliquant d'une part que son intervalle de temps était infini et d'autre part que l'échantillonnage par traversée de niveau d'un signal impulsionnel produisait un nombre fini de points. Or cette hypothèse était en un sens restrictive car elle ne tenait pas compte d'une réalité physique : la profondeur du timer. En effet, le timer est un compteur dont le cardinal est fini. Ainsi pour une largeur N_T et une résolution T_c, il est capable de mesurer un intervalle de temps dans la plage $[T_c ; N_T T_c]$. Lorsque l'intervalle de temps dépasse la durée maximale mesurable $T_F = N_T T_c$, un point est prélevé pour réinitialiser le compteur.

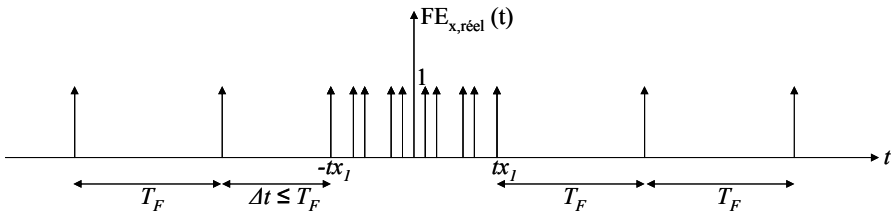


Figure 37 : Fonction d'échantillonnage d'un signal impulsionnel avec saturation du timer

En démarrant la conversion à un instant t_0 (avec t_0 tendant vers $-\infty$), une série de points est échantillonné uniformément avec un pas temporel T_F, tant que le signal ne dépasse pas le premier niveau. Puis le signal est échantillonné non uniformément en fonction des variations de l'impulsion jusqu'à ce qu'il repasse sous le premier niveau. A partir de cet instant, une nouvelle série de points espacés d'une durée T_F est prélevée. Cette fois-ci, il n'y a plus d'intervalle de temps infini. En revanche, la durée Δt entre le dernier point échantillonné automatiquement et le premier point correspondant à la traversée du premier niveau est aléatoire car elle dépend de l'instant d'initialisation t_0. Ainsi finalement, il est possible de décrire l'échantillonnage comme la multiplication entre le signal analogique et une fonction d'échantillonnage composée d'un peigne de Dirac non uniforme de longueur infinie dont une représentation est donnée sur la Figure 37.

La fonction d'échantillonnage contient trois parties distinctes : un peigne de Dirac uniforme de période T_F, puis un peigne de Dirac non uniforme dont les intervalles de temps dépendent des variations du signal et enfin un nouveau peigne de Dirac uniforme de période T_F. L'étude du schéma d'échantillonnage ne peut alors plus s'effectuer par transformée de Fourier car la fonction a des propriétés spectrales qui évoluent au cours du temps ; l'échantillonnage est non stationnaire :

$$FE_{x,réel}(t) = \begin{cases} \sum \delta\left(t + nT_F + tx_1 + \Delta t\right) & \forall t << -tx_1 \\ FE_x(t) & \forall t \in \left[-tx_1 ; tx_1\right] \\ \sum \delta\left(t - nT_F - tx_1\right) & \forall t >> tx_1 \end{cases} \qquad \text{Eq. (74)}$$

Nous devons donc procéder par une analyse temps-fréquence. Ainsi, en calculant la transformée de Fourier sur une fenêtre d'observation glissante, nous obtenons le schéma d'échantillonnage de l'impulsion représenté sur la Figure 38 :

$$SE_{x,réel}(f,t) = \begin{cases} e^{j2\pi f\left(tx_1+\Delta t\right)} \cdot \sum_{n\in\mathbf{Z}} W\left(f - \frac{n}{T_F}\right) & \forall t << -tx_1 \\ SE_x(f) & \forall t \in \left[-tx_1 ; tx_1\right] \\ e^{-j2\pi f\left(tx_1\right)} \cdot \sum_{n\in\mathbf{Z}} W\left(f - \frac{n}{T_F}\right) & \forall t >> tx_1 \end{cases} \qquad \text{Eq. (75)}$$

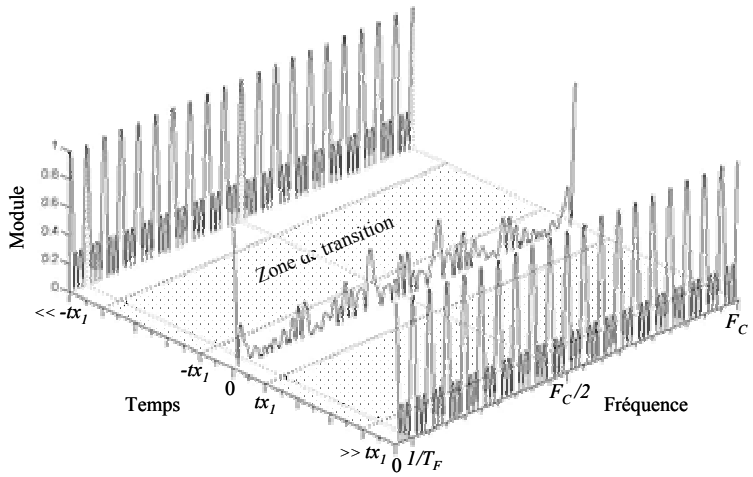


Figure 38 : Représentation temps-fréquence théorique du schéma d'échantillonnage d'un signal impulsionnel avec saturation du timer

Tant qu'il n'y a pas de variations du signal impulsionnel, le schéma d'échantillonnage est un peigne de Dirac uniforme de période $1/T_F$ *convolué* par la transformée de Fourier *W(f)* de la fenêtre temporelle *w(t)* utilisée pour l'analyse. Puis après une période de transition, nous retrouvons le schéma d'échantillonnage obtenu sans saturation du timer. Enfin lorsqu'il n'y a plus de variation, le schéma d'échantillonnage redevient un peigne de Dirac uniforme de période $1/T_F$.

En utilisant ce résultat, nous ne pouvons plus étudier le spectre du signal échantillonné de la même manière. Nous devons inclure un paramètre temporel pour utiliser une représentation temps-

fréquence de l'échantillonnage. Par ailleurs, dans le cadre d'un signal analogique impulsionnel, nous pouvons simplifier sa transformée de Fourier puisque le signal a une durée utile $T_x = 2tx_1$:

$$\begin{cases} X(f) \approx \int_{-tx_1}^{tx_1} x(t) e^{-j2\pi ft} dt \\ x(t)\big|_{|t|>tx_1} \approx 0 \end{cases} \qquad \text{Eq. (76)}$$

Ceci nous permet d'obtenir directement, aux transitions près, la représentation temps-fréquence du signal analogique :

$$\begin{cases} X(f,0) \approx X(f) \\ X(f,|t| > tx_1) \approx 0 \end{cases} \qquad \text{Eq. (77)}$$

Finalement, nous pouvons en déduire la représentation temps-fréquence du signal échantillonné avec saturation du timer : nous conservons en $t = 0$ le même spectre que le signal échantillonné sans saturation tandis qu'en $|t| > tx_1$ nous obtenons un spectre nul. Ce résultat permet alors de simplifier le paramètre temporel puisqu'il n'y a qu'un seul instant pour lequel la représentation n'est pas nulle.

En conclusion, la saturation du timer qui force le convertisseur à échantillonner un point pour réinitialiser le compteur du timer induit localement un schéma d'échantillonnage composé d'impulsions de Dirac très brèves espacées de $1/T_F$. Toutefois, ce schéma n'a aucune influence car il se produit pendant une période où il n'y a aucune activité, donc où le contenu spectral du signal analogique est nul. La saturation du timer ne modifie donc pas le schéma d'échantillonnage initial ; il permet juste de faire *une pause* pendant l'absence de toute variation significative du signal.

III.1.4 Extension aux signaux impulsionnels périodiques

Un cas intéressant se produit lorsque plusieurs impulsions espacées d'une durée T_o se succèdent, comme dans le cas des signaux d'électrocardiogrammes par exemple. En effet, le signal est alors à la fois localement impulsionnel imposant la saturation du timer et globalement périodique donc décomposable sur son ensemble en série de Fourier. De ce simple constat, nous pouvons donc en déduire deux approches possibles pour l'étude de ce genre de signaux.

La première approche consiste à utiliser la périodicité du signal pour simplifier l'étude à un signal périodique comme il l'a été montré dans la section II.2. Dans ce cas, la fonction

d'échantillonnage est périodique de période T_o et le schéma d'échantillonnage est une somme raies de fréquence fondamentale $f_o = 1/T_o$. Le spectre du signal échantillonné est alors lui-même un spectre de raies de fréquence fondamentale f_o. De plus, l'amplitude d'une raie à une fréquence donnée est égale à celle du spectre du signal non périodique échantillonné. Néanmoins, cette approche souffre d'un inconvénient : nous serons limités lors de l'étude des signaux réels étant donné que les impulsions doivent être rigoureusement identiques, c'est-à-dire que le motif de base doit toujours rester le même.

Dans cette perspective, une seconde approche est déduite de la section précédente : effectuer l'analyse temps-fréquence du signal échantillonné. Ainsi quelle que soit la période T_o et donc quelle que soit la durée de saturation, il est possible d'étudier l'échantillonnage à condition que T_o soit beaucoup plus grande que T_F afin d'assurer de grandes périodes spectrales nulles pour observer facilement les zones de transitions.

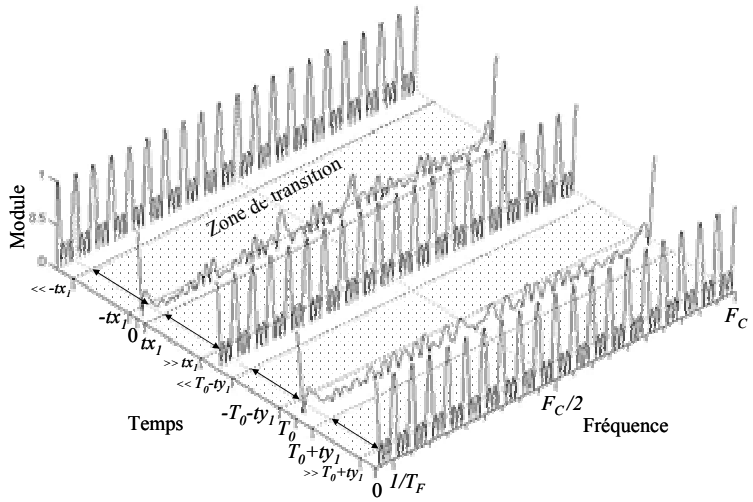


Figure 39 : Représentation temps-fréquence théorique du schéma d'échantillonnage d'un signal impulsionnel variable de période T_0 avec saturation du timer

Dans l'exemple présenté Figure 39, la largeur des impulsions varie au cours du temps. L'analyse temps fréquence nous permet de dissocier les impulsions les unes des autres. Ainsi, nous pouvons observer deux schémas d'échantillonnage distincts pour la première impulsion, centrée en 0, appelée x (définie entre $-tx_1$ et tx_1) et pour la deuxième, centrée en T_o, appelée y (définie entre $T_o - ty_1$ et $T_o + ty_1$).

Pour illustrer ces passages successifs du schéma d'échantillonnage entre peigne de Dirac régulier et non uniforme, nous avons simulé un signal composé de deux impulsions distantes d'une

durée telle que le timer arrive à saturation plusieurs fois. La première, notée *x*, a une durée utile de 2,83s ($\alpha = 1$) et est centrée à l'instant $t = 60s$ tandis que la seconde, notée *y*, a une durée de 8,94s ($\alpha = 0,1$) et est centrée à l'instant $t = 120s$. L'échantillonnage est réalisé sur 15 niveaux et la quantification du temps est effectué par un timer de fréquence $F_C = 20Hz$ mesurant un intervalle de temps maximum $T_F = 5s$ (Figure 40)

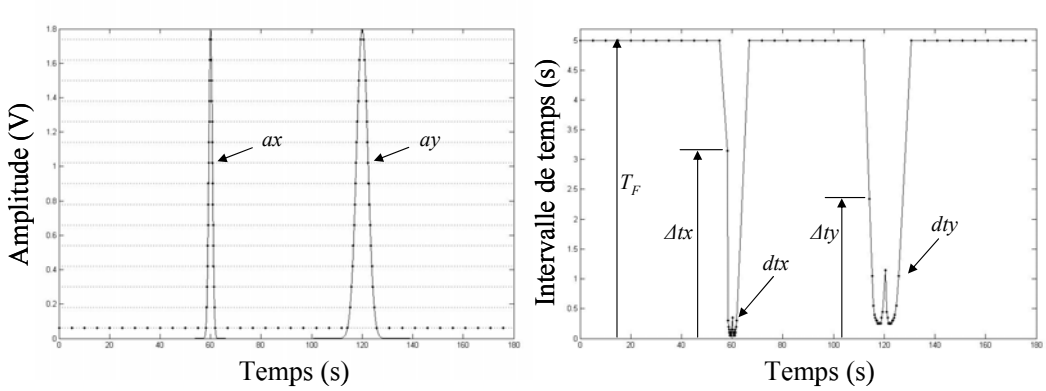


Figure 40 : Echantillonnage par traversée de niveaux de deux impulsions décalées. Des périodes inactives induisent la saturant du timer.

Nous avons analysé le schéma d'échantillonnage entre 0 et 1Hz à l'aide d'une fenêtre rectangulaire de largeur $T_W = 20s$ pour réaliser une représentation à bande étroite (partie gauche de la Figure 41). Nous pouvons ainsi aisément observer les alternances entre un peigne de Dirac de période $1/T_F = 0,2Hz$ lorsque le signal est inactif et le schéma d'échantillonnage d'une impulsion lorsque que l'amplitude du signal évolue. Les transitions entre les deux modes sont clairement définies en fréquence; un zoom autour de la première impulsion est présenté Figure 42 montrant le passage d'un schéma d'échantillonnage continu en fréquence à un schéma discrétisé. En revanche, à cause de l'incertitude temps-fréquence, il n'est pas possible de déterminer précisément, sur la représentation à bande étroite, l'apparition des impulsions : toutes deux sont en effet analysées dans une région égale à la durée de la fenêtre ($T_W = 20s$) et leurs largeurs respectives ne sont donc pas discernables. En considérant une fenêtre de largeur $T_W = 5s$, nous réalisons alors une représentation large bande pour améliorer la résolution temporelle (partie gauche de la Figure 41). Le peigne de Dirac en fréquence n'est plus affiché du fait de la dégradation de la résolution fréquentielle. Toutefois les impulsions sont cette fois-ci clairement différenciables même si les zones de transition ne sont plus définies aussi précisément.

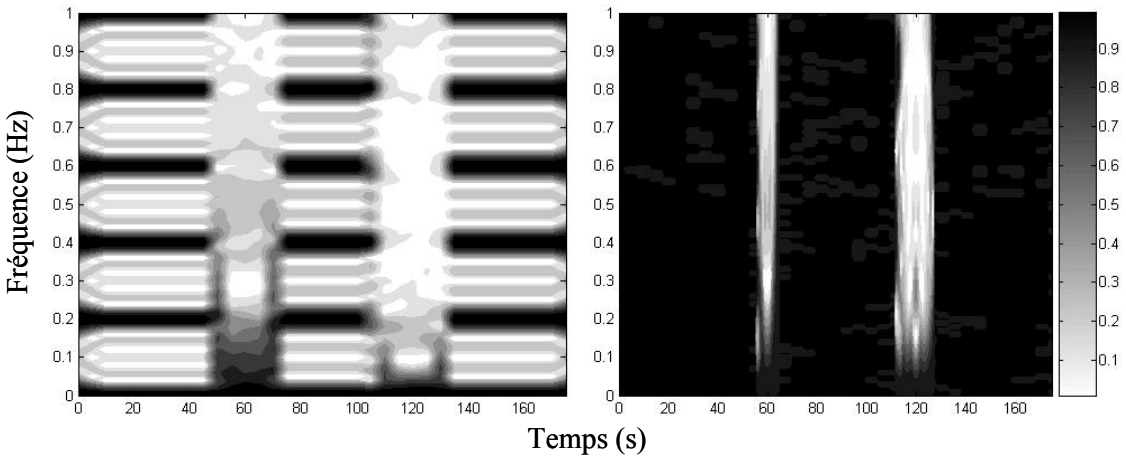


Figure 41 : Représentations temps fréquence à bande étroite (à gauche) et large bande (à droite) en lignes de niveaux du schéma d'échantillonnage de deux impulsions décalées

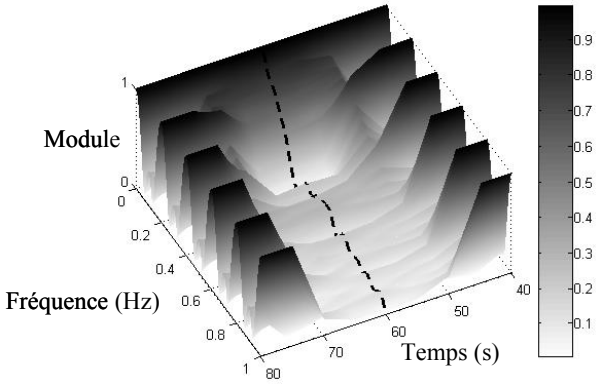


Figure 42 : Transition entre le peigne de Dirac en fréquence et le schéma d'échantillonnage d'une impulsion

Nous pouvons ainsi généraliser ce résultat à tout type de signaux possédant de longues périodes d'inactivité en préférant une analyse temps-fréquence à bande étroite et large bande pour distinguer les différentes phases du signal et leurs schémas respectifs.

III.2 Etude d'un signal non stationnaire

Après avoir étudié différentes formes de signaux pour lesquels il était possible de comprendre le schéma d'échantillonnage à partir des intervalles de temps, nous proposons de généraliser ce que nous venons d'expliquer sur un signal non périodique et non impulsionnel. Afin

de simplifier l'étude de son schéma d'échantillonnage, nous nous limiterons à une modulation linéaire de fréquence : en effet, comme ce signal est localement périodique, il nous permettra de nous rattacher à la théorie précédente sur les signaux périodiques.

Pour ce genre de signaux, il est possible d'effectuer une transformée de Fourier mais cette représentation n'apporte aucune information sur les propriétés spectrales étant donné qu'elles évoluent au cours du temps. Il est alors préférable d'utiliser à nouveau une représentation temps-fréquence afin d'observer la fréquence instantanée du signal.

Pour une modulation linéaire de fréquence de durée T et d'accroissement A, le signal est défini en fonction de la fréquence instantanée :

$$\begin{cases} f_i(t) = f_0 + Atu(t) - A(t-T)u(t-T) \\ x(t) = a_M + a_0 \sin(2\pi f_i(t)) \end{cases}, \qquad \text{Eq. (78)}$$

où $u(t)$ est l'échelon de Heavyside. La fréquence instantanée est représentée sur la Figure 43.

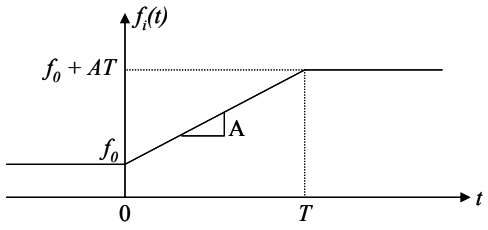


Figure 43 : Fréquence instantanée d'une modulation linéaire de fréquence

III.2.1.1 Schéma d'échantillonnage

Comme nous l'avons vu précédemment, un signal sinusoïdal de fréquence f_o a un schéma d'échantillonnage formé d'une somme d'impulsions de Dirac de fréquence fondamentale f_o dont les amplitudes ne dépendent, pour un convertisseur donné, que de l'amplitude du signal.

Dans le cadre d'une modulation linéaire de fréquence, la fréquence de la sinusoïde évolue au cours du temps. Ainsi le schéma d'échantillonnage et le spectre du signal échantillonné évoluent également de la même manière : les impulsions de Dirac (*convolués* par la transformée de Fourier de la fenêtre d'observation $W(f)$) conservent leurs amplitudes (aux effets de fuites d'énergie près) mais sont linéairement dilatées en fréquence.

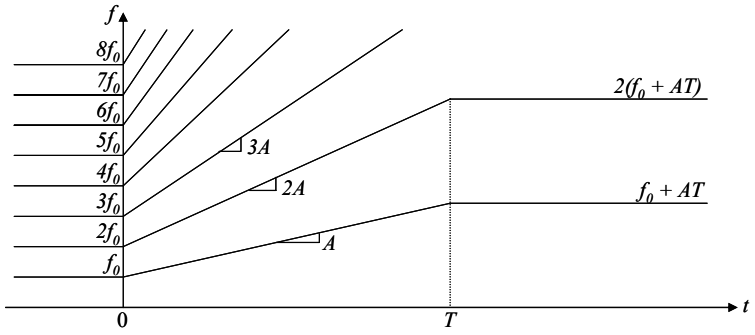


Figure 44 : Allure théorique du schéma d'échantillonnage et du spectre d'une modulation linéaire de fréquence échantillonnée

III.2.1.2 Exemple de représentation temps-fréquence d'un signal non périodique

En accord avec le schéma d'échantillonnage, la représentation temps-fréquence nous permet d'étudier les variations du spectre du signal au cours du temps. A la différence du signal impulsionnel où le calcul du spectre peut être limité à la durée utile, un signal non périodique et non impulsionnel évolue continuellement au cours du temps. Nous proposons donc de simuler une modulation linéaire de fréquence instantanée telle que nous l'avons définie à travers l'équation (78) et représentée sur la Figure 45.

L'échantillonnage par traversée de niveaux est effectué à l'aide d'un convertisseur 4 bits c'est-à-dire d'un convertisseur disposant de 15 niveaux répartis uniformément dans la dynamique $[0 ; V_{alim} = 1.8V]$. Le signal analogique est un signal sinusoïdal de valeur moyenne $a_M = 0.86V$ (afin de s'ajuster dans une position dissymétrique) et d'amplitude crête $a_o = 0.86V$ (pour obtenir une plage de variation maximum dans la dynamique du convertisseur). La fréquence instantanée est initialisée à $f_o = 1Hz$ puis augmente linéairement à partir d'un instant $t_1 = 5s$ pendant une durée $T = 10s$ avec une pente de $A = 0.9Hz/s$. Enfin la fréquence se stabilise jusqu'au terme de la simulation à $f_o + AT = 10Hz$.

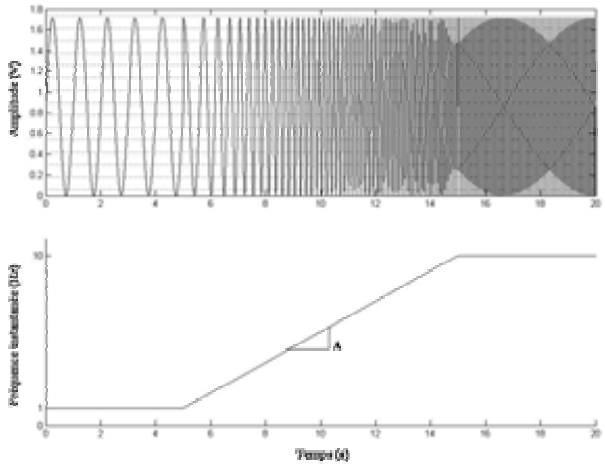


Figure 45 : Exemple de modulation linéaire de fréquence

En fixant une fenêtre rectangulaire de largeur $T_W = 1s$ et en calculant la transformée de Fourier glissante (avec des sauts 0,25s) de la fonction d'échantillonnage et du signal échantillonné non uniformément, nous obtenons respectivement les deux représentations de la Figure 46. A première vue, comme nous pouvions nous y attendre, les résultats sont assez similaires : le schéma d'échantillonnage et le spectre sont composés d'un train d'impulsions et de fréquence fondamentale la fréquence instantanée du signal, dont les amplitudes dépendent des variations du signal. L'amplitude de chaque harmonique reste constante au cours du temps du fait de son indépendance avec la fréquence fondamentale. Elle dépend au contraire de la valeur moyenne du signal et son amplitude crête ainsi que du convertisseur.

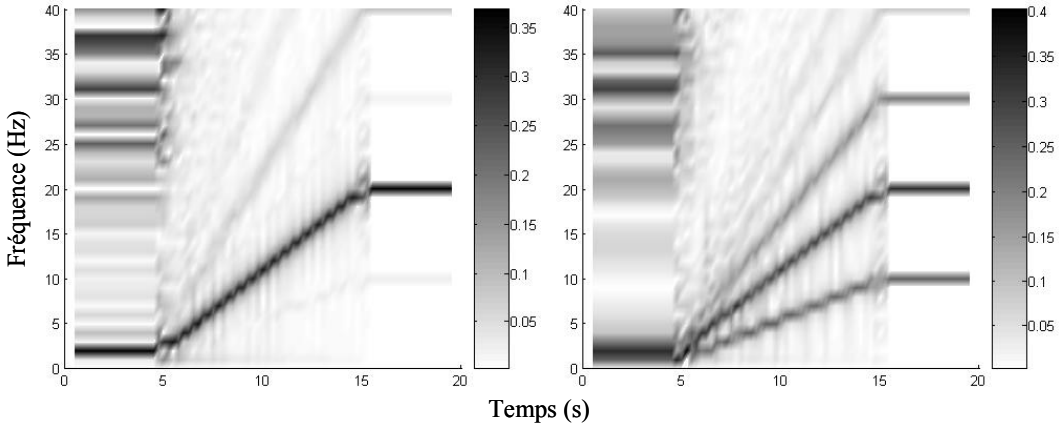


Figure 46 : Représentation temps-fréquence du schéma d'échantillonnage et du spectre du signal échantillonné dans le cadre d'une modulation linéaire de fréquence

III.3 Conclusion : l'échantillonnage et le repliement de spectre

Nous venons d'étudier dans ce chapitre et dans le précédent, l'échantillonnage par traversée de niveaux de quatre types de signaux. Bien qu'il s'agisse là d'une liste non exhaustive de situations (dans la mesure où l'échantillonnage dépend directement du signal analogique) ils représentent dans leur ensemble tous les cas de figure que nous pouvons rencontrer.

L'étude analytique de ces signaux nous a permis de montrer que la fonction d'échantillonnage qui réalise mathématiquement l'opération de prélèvement, dépend du signal et plus précisément de son évolution. Ceci implique que les propriétés statistiques de cette fonction dépendent des propriétés statistiques des variations du signal (donc des amplitudes et du temps). De ce fait, la fonction d'échantillonnage dépend directement de la fréquence instantanée du signal. Ainsi, pendant des zones actives, un nombre important de points sera échantillonné tandis que pendant des périodes de *repos*, peu de points voire aucun ne seront prélevés. Cela a un avantage certain : il permet de compresser dynamiquement et *en temps réel* le signal lors de son échantillonnage. L'inconvénient est que le spectre du signal échantillonné, obtenu par transformée de Fourier (i.e. par la GDFT), est systématiquement replié. Aucune information sur le signal analogique ne peut être déduite à partir du spectre du signal échantillonné. Ceci vient du fait que la GDFT ne tient pas compte de la distance entre deux points car elle ne traite que l'amplitude des points, leur donnant du coup la même valeur informative. Or, avec un échantillonnage par traversée de niveaux, chaque intervalle de temps contient une information importante sur le signal. En effet, si l'intervalle de temps est grand par exemple, cela veut dire qu'il y a peu de points dans son entourage et qu'il est donc singulier. Bien qu'elle ne nous intéresse pas en terme de donnée à traiter, l'information contenue dans cet intervalle est significative : *« L'amplitude du signal n'a pas évoluée de plus d'un quantum depuis très longtemps ».* Lors de variations du signal analogique, les intervalles sont plus petits et de même grandeurs. Dans cette configuration, l'information des intervalles de temps permet de pondérer chaque amplitude.

Par conséquent, pour traiter un signal échantillonné par traversée de niveaux, il faut concevoir des algorithmes traitant explicitement la valeur des intervalles de temps en plus des amplitudes.

CHAPITRE IV

Reconstruction des signaux échantillonnés non uniformément

Dans ce chapitre, nous abordons la reconstruction d'un signal à temps continu à partir d'un signal à temps discret échantillonné non uniformément.

Comme nous l'avons évoqué dans le premier chapitre, la plupart des signaux sont naturellement échantillonnés non uniformément. La majorité des publications qui leur sont dédiées, traitent de la reconstruction idéale et sont donc assez difficiles à synthétiser. Il est intéressant de noter que parmi l'ensemble des ouvrages consultés, aucune étude ne tient compte la spécificité particulière de l'échantillonnage par traversée de niveaux. Nous ne proposerons donc qu'une vue d'ensemble des principaux résultats classés chronologiquement puisque seule une reconstruction pratique, c'est-à-dire implémentable, nous intéresse.

Dans la suite, nous étudions la reconstruction par bloqueur d'ordre 0 et d'ordre 1. Nous montrons, à partir du spectre du signal reconstruit, que la perte d'information, mise en évidence dans les chapitres précédents par un repliement de spectre systématique du signal échantillonné, pouvait être compensée par l'utilisation de l'information contenue dans les intervalles de temps.

Par ailleurs, nous montrons que le spectre du signal reconstruit par bloqueur d'ordre 0 et d'ordre 1 ne dépend que de l'amplitude et de l'intervalle de temps des échantillons. Il est donc possible d'utiliser une approximation numérique du spectre du signal reconstruit à la place de la GDFT pour réaliser une analyse spectrale du signal échantillonné.

IV.1 Reconstruction d'un signal à temps continu

IV.1.1 Reconstruction idéale

La reconstruction théorique d'un signal échantillonné non uniformément fait appel à des méthodes très complexes. Elle rassemble deux aspects distincts : la reconstruction du signal à temps continu initial et le ré-échantillonnage du signal régulièrement dans le temps. Bien que la reconstruction idéale ne fasse pas partie de nos objectifs, nous proposons de résumer, de manière non exhaustive, l'historique de ce travail.

L'un des premiers travaux fut celui de Paley et Wiener [Paley *et al.* 1934]. Ils ont montré qu'un signal à bande limitée de fréquence maximale F_{max} échantillonnée irrégulièrement à des instants connus $\{t_n\}$ pouvait être reconstruit grâce à un ensemble de fonctions $\{g_n(t)\}$ selon :

$$x(t) = \sum_{n \in \mathbf{Z}} x(t_n) g_n(t) \qquad \text{Eq. (79)}$$

si les instants respectaient la condition suivante : $\sup_{n \in \mathbf{Z}} \left| t_n - \frac{n}{2F_{max}} \right| \leq L < \infty$. Dans le cas de l'échantillonnage régulier, les fonctions g_n sont directement des sinus cardinaux (obtenus par transformée de Fourier Inverse d'un filtre passe bas idéal). En 1940, Levinson [Levinson 1940] donne une formulation des fonctions g_n dans le cas général basée sur l'interpolation de Lagrange :

$$g_n(t) = \frac{g(t)}{g'(t)(t - t_n)} \text{ avec } g(t) = (t - t_0) \prod_{n \in \mathbf{Z}^*} \left(1 - \frac{t}{t_n} \right) \qquad \text{Eq. (80)}$$

Malheureusement l'utilisation directe de cette formule est très complexe rendant presque impossible toute implémentation matérielle.

Puis en 1952 est introduit la théorie des frames. Duffin et Schaeffer [Duffin *et al.* 1952] définissent un frame comme un ensemble de fonctions $\{g_n\}$ appartenant à un espace de Hilbert H (introduisant le produit scalaire noté $\langle .,. \rangle$) satisfaisant la relation suivante :

$$A\|x\|^2 \leq \sum_{n \in \mathbf{Z}} \left| \langle x, g_n \rangle \right|^2 \leq B\|x\|^2, \ \forall x \in H \qquad \text{Eq. (81)}$$

où A et B sont deux constantes strictement positives à déterminer. Le détail du formalisme mathématique associé aux frames est présenté par Werther [Werther 1999]. Duffin et Schaeffer ont alors montré que s'il existe trois paramètres $L>1$, α et $0<\varepsilon<1$ tels que tous les instants d'échantillonnage vérifient les conditions $|t_n - t_m| \geq \alpha$ pour $n \neq m$ et $\sup_{n\in\mathbf{Z}} \left| t_n - \frac{n\varepsilon}{2F_{max}} \right| \leq L < \infty$ alors les deux constantes A et B peuvent être déterminées en fonction de L, α et ε :

$$A\|x\|^2 \leq \sum_{n\in\mathbf{Z}} |x(t_n)|^2 \leq B\|x\|^2 \qquad \text{Eq. (82)}$$

Le paramètre le plus important L a été borné en 1964 par Kadec [Kadec 1964]. Il a en effet montré qu'il est possible de reconstruire le signal si les instants d'échantillonnage irréguliers ne sont pas déviés de plus de 25% du point idéal, c'est-à-dire en posant $L = 0{,}25/(2F_{max})$. Dans la littérature, ce résultat est présenté comme le théorème de Kadec 1/4. Finalement le théorème de Paley-Wiener et de Kadec a été reformulé maintes fois notamment par Beutler [Beutler 1966] et Yao [Yao *et al.* 1967] pour tendre vers une relation simple généralisant le théorème de Shannon :

$$\overline{F}_e = \lim_{N\to+\infty} \left(\frac{2N+1}{\sum_{-N}^{+N} dt_n} \right) \geq 2F_{max} \qquad \text{Eq. (83)}$$

Le signal peut être reconstruit si la fréquence d'échantillonnage moyenne est supérieure ou égale au double de la fréquence maximale du signal [Jerri 1977]. Appelé communément théorème de Beutler, ce résultat introduit la notion de densité d'échantillonnage qui avait permis à Nyquist d'anticiper le théorème de Shannon. Si l'échantillonnage est suffisamment dense, le signal sera reconstructible.

Feichtinger et Gröchenig [Feichtinger *et al.* 1994] (puis globalement les travaux du groupe de recherche NuHag de l'université de Vienne [NuHag]) ont alors proposé de généraliser la théorie de l'échantillonnage à partir des frames. Pour une séquence d'instants d'échantillonnage $\{t_n\}$ vérifiant la condition $|t_n - t_m| \geq \alpha > 0$ pour $n \neq m$, alors l'ensemble de fonctions $\{T_{t_n} sinc_{f_{max}}\}$ (où T est l'opérateur de translation) est un frame si la relation (83) est vérifiée. La réciproque est également vraie. Ainsi pour reconstruire le signal, ils introduisent l'opérateur de frame S défini par $Sx = \sum_{n\in\mathbf{Z}} \langle x, g_n \rangle g_n$ leur permettant de définir une série :

$$x^{(0)} = Sx \text{ et } x^{(m)} = x^{(m-1)} + \lambda S\left(x - x^{(m-1)}\right) \text{ pour m} > 0 \qquad \text{Eq. (84)}$$

Au départ, le signal est reconstruit *grossièrement* à partir des échantillons $x(t_n)$ sur la base des fonctions g_n du frame puis l'erreur est intégrée successivement. Au final le signal d'origine est reconstruit : $x = \lim_{m \to +\infty} x^{(m)}$ avec une erreur bornée $\left\| x - x^{(m)} \right\| \le \gamma^{m+1} \|x\|$. Cependant les auteurs admettent que l'application de ce théorème est limitée par l'absence d'estimateurs pour le calcul des bornes A et B d'autant plus qu'ils influencent les performances en terme de stabilité et de convergence :

$$\lambda = \frac{2}{A+B} \text{ et } \gamma = \frac{B-A}{B+A} \qquad \text{Eq. (85)}$$

Ils proposent donc des algorithmes itératifs de reconstruction où chaque méthode est étudiée en fonction du paramètre de relaxation λ et du taux de convergence γ. On peut citer notamment la méthode des poids adaptés basée sur des frames pondérés où les bornes A et B ne dépendent uniquement que de l'intervalle de temps le plus grand et de la fréquence maximale du signal :

$$\text{Si } \delta = \sup_{n \in \mathbf{Z}} (t_{n+1} - t_n) < \frac{1}{2f_{max}} \text{ alors } A = \left(1 - \frac{\delta}{2f_{max}}\right) \text{ et } B = \left(1 + \frac{\delta}{2f_{max}}\right) \qquad \text{Eq. (86)}$$

Parallèlement, Papoulis [Papoulis 1977] a démontré qu'un signal à bande limitée peut être reconstruit à partir des échantillons de m systèmes linéaires et invariant dans le temps échantillonnés à $1/m$ fois la fréquence de Nyquist (bien que localement chaque système soit sous-échantillonné, la fréquence d'échantillonnage moyenne de l'ensemble des systèmes réunis est égale à la fréquence de Nyquist). En généralisant les travaux de Papoulis, Eldar [Eldar *et al.* 2000] est parvenue à implémenter la relation de Levinson. Ainsi, dans le cas particulier d'un échantillonnage non uniforme périodique, elle propose d'utiliser un banc de filtres à temps continu pour reconstruire le signal ou à temps discret pour interpoler les échantillons.

Par ailleurs, dans le cadre des signaux multi-bandes c'est-à-dire dont le contenu spectral est défini comme l'union U d'intervalles de bandes fréquentielles B, Landau [Landau 1967] a généralisé le théorème de Beutler en montrant que pour une union U connue, le signal peut être reconstruit si la fréquence d'échantillonnage moyenne est plus grande que B. Venkataramani *et al.* proposent sur ce principe plusieurs techniques de reconstruction de signaux multi-bandes [Venkataramani *et al.* 2000] [Venkataramani *et al.* 2001].

Enfin, dans le cadre de l'échantillonnage non uniforme par traversée de niveaux, aucun travail ne s'est intéressé à la reconstruction théorique du signal en tenant compte de sa spécificité. En revanche lors la conception du convertisseur [Allier 2003], le théorème de Beutler a été pris en compte pour fixer le nombre de niveaux afin d'assurer une densité d'échantillonnage suffisamment élevée en vue de la reconstruction. Cependant, il s'avère que le théorème de Beutler, tel qu'il est présenté par l'équation (83) est limitant : en effet, étant donné que le processus d'échantillonnage n'est plus forcément stationnaire, on ne doit plus considérer la fréquence d'échantillonnage moyenne totale mais introduire une moyenne locale ou glissante. Si l'échantillonnage est stationnaire, dans le cadre d'un jitter par exemple, la moyenne des fréquences d'échantillonnage n'évolue pas au cours du temps; il est donc possible de la comparer à la fréquence maximale du signal même si celui-ci n'est pas stationnaire en prenant en compte la plus grande des fréquences maximales. Mais si l'échantillonnage est adapté aux variations du signal, les fréquences d'échantillonnage vont s'adapter aux variations du contenu spectral du signal en particulier s'il est non stationnaire; la moyenne des fréquences d'échantillonnage évolue donc au cours du temps. Or, une moyenne globale peut ne pas respecter le théorème de Beutler bien qu'il le soit localement en particulier si une longue période inactive (c'est-à-dire un contenu basse fréquence) diminue la moyenne comme le montre la Figure 47. Sur la partie gauche de la figure, la fréquence d'échantillonnage moyenne est deux fois plus grande que la fréquence maximale du signal; le théorème de Beutler est respecté alors que sur la partie droite, la fréquence moyenne n'est pas deux fois plus grande bien que elle le soit localement. Finalement il faut donc considérer le théorème de Beutler localisé dans un intervalle temporel : $\overline{F_e}(t) \geq 2F_{max}(t)$, où $\overline{F_e}(t)$ représente la fréquence moyenne d'échantillonnage sur l'intervalle temporel considéré et $F_{max}(t)$ représente la fréquence maximale du signal à temps continu dans le même intervalle.. A partir de ce point nous n'avons pas poursuivi l'étude de la reconstruction théorique car elle ne faisait partie des objectifs de base. Toutefois, nous pensons qu'en segmentant les régions actives et inactives, le signal pourrait être reconstruit par morceaux à l'aide des techniques énoncées précédemment malgré l'incertitude temps-fréquence amenée par la localisation (détermination de $\overline{F_e}(t)$ et de $F_{max}(t)$) et les problèmes de continuité entre les morceaux reconstruits.

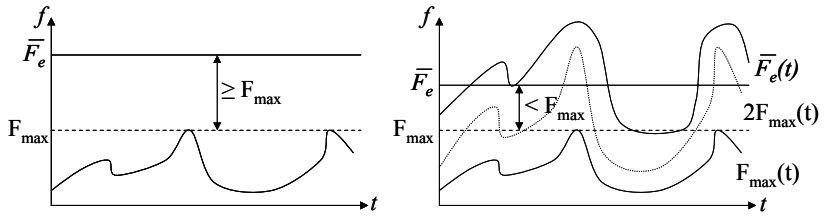


Figure 47 : Interprétation du théorème de Beutler dans le cadre de l'échantillonnage stationnaire (figure de gauche) et non stationnaire (figure de droite) d'un signal non stationnaire

IV.1.2 Reconstruction pratique

Comme nous l'avons précisé dans l'introduction, le but premier de ce travail est de repenser les chaînes de traitement numérique du signal. La reconstruction d'un signal à temps continu s'effectue systématiquement par un convertisseur numérique-analogique utilisant, pour des raisons de complexité matérielle, un bloqueur d'ordre 0 voire parfois d'ordre 1. A ce sujet, De Waele [De Waele *et al.* 2000] a comparé plusieurs techniques d'interpolation pour le ré-échantillonnage de donnée : d'un coté des méthodes *simples* (bloqueur d'ordre 0 et interpolation par le voisin le plus proche, n'utilisant qu'un seul échantillon dans le calcul) et de l'autre des méthodes *complexes* (interpolation d'ordre 1 et splines cubiques, utilisant plusieurs échantillons). Il montre que les méthodes simples d'interpolation sont plus robustes car elles n'entraînent jamais l'apparition d'oscillations erronées. Ses résultats vont dans le sens des contraintes d'implémentation imposant un bloqueur d'ordre 0 et nous amènent donc à choisir ce type d'interpolation pour la reconstruction d'un signal à temps continu [Aeschlimann *et al.* 2005].

IV.1.2.1 Reconstruction par bloqueur d'ordre 0

Avec un bloqueur d'ordre 0, le signal reconstruit est obtenu par convolution entre le signal échantillonné $x_E(t)$ (équation (2)) et des fonctions rectangulaires dont les largeurs dépendent des intervalles de temps :

$$x_0(t) = x_E(t) * \underset{\theta}{\Pi}(t) \qquad \text{Eq. (87)}$$

Une fonction rectangulaire est définie pour n'importe quel réel θ par :

$$\Pi_{\theta}(t) = \begin{cases} 1 & \text{pour } t \in \left[-\frac{\theta}{2}; \frac{\theta}{2} \right[\\ 0 & \text{ailleurs} \end{cases} \qquad \text{Eq. (88)}$$

En centrant chaque fonction porte entre les instants tx_n et tx_{n-1}, le signal reconstruit peut alors s'écrire :

$$x_0(t) = \sum_{n \in \mathbf{Z}} ax_n \, \Pi_{dtx_n}\left(t - tx_n + \frac{dtx_n}{2} \right) \qquad \text{Eq. (89)}$$

Puis grâce aux propriétés de linéarité et de translation de la transformée de Fourier, le spectre du signal reconstruit est déduit de l'équation précédente :

$$X_0(f) = \lim_{T \to +\infty} \frac{1}{T} \sum_{n=\frac{-N_T}{2}}^{\frac{N_T}{2}-1} ax_n W_n^0(f) e^{-j2\pi f tx_n} \;,\; \forall f \in \mathbf{R} \qquad \text{Eq. (90)}$$

où N_T est le nombre de fonctions rectangulaires dans une fenêtre de largeur T et $W_n^0(f)$ représente la pondération à l'ordre 0 du $n^{ième}$ échantillon :

$$W_n^0(f) = dtx_n e^{-j\pi f dtx_n} sinc(\pi f dtx_n) \qquad \text{Eq. (91)}$$

En simulation, le signal échantillonné est tronqué par une fenêtre rectangulaire de largeur T fixée permettant le traitement d'une séquence de N points consécutifs sur une série de fréquences particulières f :

$$\widetilde{X}_0(f) = \frac{1}{T} \sum_{n=0}^{N-1} ax_n W_n^0(f) e^{-j2\pi f tx_n} \qquad \text{Eq. (92)}$$

IV.1.2.2 Reconstruction par interpolation d'ordre 1

Avec une interpolation d'ordre 1, le signal reconstruit est obtenu cette fois-ci par convolution entre le signal échantillonné $x_E(t)$ et des fonctions triangulaires dont les pentes dépendent des intervalles de temps :

$$x_1(t) = x_E(t) * \Delta_{\theta,\sigma}(t) \qquad \text{Eq. (93)}$$

Les fonctions triangulaires sont définies pour tous paramètres θ et σ par :

$$\underset{\theta,\sigma}{\Delta}(t)=\begin{cases}1+\dfrac{t}{\theta} & \text{pour } t\in[-\theta;0[\\ 1-\dfrac{t}{\sigma} & \text{pour } t\in[0;\sigma[\\ 0 & \text{ailleurs}\end{cases} \qquad \text{Eq. (94)}$$

Ainsi le signal reconstruit $x_1(t)$ s'écrit simplement :

$$x_1(t)=\sum_{n\in\mathbf{Z}} ax_n \underset{dtx_{n+1},dtx_n}{\Delta}(t-tx_n) \qquad \text{Eq. (95)}$$

Dans le domaine fréquentiel, le spectre du signal reconstruit sera donc obtenu par linéarité :

$$X_1(f)=\lim_{T\to+\infty}\frac{1}{T}\sum_{n=\frac{-N_T}{2}}^{\frac{N_T}{2}-1} ax_n W_n^1(f)e^{-j2\pi ftx_n},\ \forall f\in\mathbf{R} \qquad \text{Eq. (96)}$$

où $W_n^1(f)$ représente cette fois-ci le poids à l'ordre 1 du $n^{ième}$ échantillons :

$$W_n^1(f)=\frac{e^{j\pi fdtx_n}sinc(\pi fdtx_n)}{j2\pi f}-\frac{e^{-j\pi fdtx_{n+1}}sinc(\pi fdtx_{n+1})}{j2\pi f},\ \forall f\in\mathbf{R}^* \qquad \text{Eq. (97)}$$

$$\text{et } W_n^1(f)=\frac{dtx_n+dtx_{n+1}}{2} \text{ pour } f=0 \qquad \text{Eq. (98)}$$

Conjointement à l'équation (92), il est possible d'étudier en simulation le spectre du signal reconstruit à l'ordre 1 en utilisant la relation approchée :

$$\widetilde{X}_1(f)=\frac{1}{T}\sum_{n=0}^{N-1} ax_n W_n^1(f)e^{-j2\pi ftx_n} \qquad \text{Eq. (99)}$$

Il est important de noter qu'indépendamment de l'aspect mathématique de l'interpolation d'ordre 1, elle pose un problème de causalité. En effet, pour reconstruire le signal entre des instants tx_{n-1} et tx_n, il faut attendre que le point à l'instant tx_n soit arrivé pour calculer la pente à partir de des

amplitudes ax_{n-1} et ax_n et de l'intervalle de temps dtx_n puis attendre la même durée pour interpoler le signal entre tx_{n-1} et tx_n. Dans le cas régulier, comme tous les intervalles de temps sont égaux, la durée d'attente est égale à la durée d'interpolation. Les deux opérations peuvent donc être effectuées en parallèle (Figure 48a). Mais dans le cas non uniforme, si le point suivant à l'instant tx_{n+1} n'est pas encore arrivé à la fin de l'interpolation i.e. $dt_{n+1} >> dt_n$, il y a un temps mort τ_M dans l'interpolation du signal car le bloqueur ne peut pas encore connaître la prochaine pente à appliquer. Le signal à temps continu est donc maintenu constant pendant la durée entraînant une déformation. Comme on peut le voir sur la Figure 48, les points suivants sont retardés. Par ailleurs, comme la durée d'interpolation entre t_n et t_{n+1} est grande, il faut également que le système de conversion soit capable de mémoriser les points suivants (sur la Figure 48b $dt_{n+1} > dt_{n+2} + dt_{n+3}$) et doit donc intégrer une file (FIFO) pour stocker les pentes que le bloqueur doit appliquer.

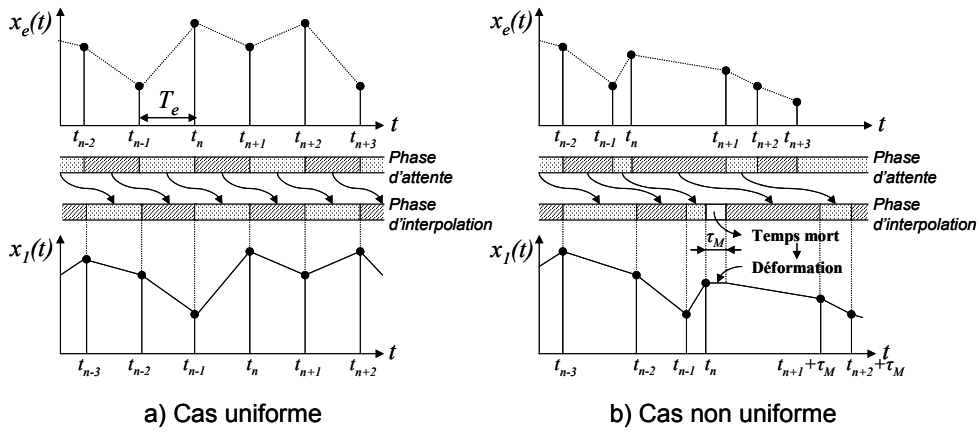


Figure 48 : Reconstruction par bloqueur d'ordre 1 dans le cas uniforme (a) et non uniforme (b)

La réalisation pratique d'un convertisseur numérique-analogique, basé sur un bloqueur d'ordre 1, entraînerait des retards donc des déformations du signal reconstruit dès qu'une durée trop importante entre deux points successifs permettrait à la file de se vider complètement.

IV.2 Etude de la distorsion du signal reconstruit

IV.2.1 Analyse spectrale : comparaison avec la GDFT

Bien que le spectre du signal reconstruit soit calculé par une transformée de Fourier Continue, il ne dépend que des valeurs des échantillons (amplitudes, intervalles de temps) et peut donc être échantillonné en fréquence comme le montrent les équations (92) et (99). Conformément

aux remarques présentées dans la conclusion du chapitre précédent, elles rendent possible l'analyse du spectre du signal à temps continu à partir des échantillons par d'autres méthodes que la transformée de Fourier Généralisée : en pondérant l'amplitude de chaque point par une fonction dépendant des intervalles de temps, elles transforment le repliement par une distorsion.

Un exemple simple est présenté sur la Figure 49. Le signal à temps continu est une sinusoïde de fréquence 1Hz, échantillonnée non uniformément sur 15 niveaux. A gauche, le spectre obtenu par GDFT montre comme dans le chapitre précédent que le repliement empêche toute analyse pertinente du signal d'entrée (erreur sur le fondamental de 4,2dB et 1er pic parasite supérieur de 1,5dB). A droite la transformée de Fourier du signal reconstruit à l'ordre 0 permet d'établir le contenu spectral du signal d'entrée (erreur sur le fondamental de 0,2dB et 1er pic parasite inférieur de 37,7dB).

Nous faisons ici l'hypothèse que le signal d'entrée n'est pas un processus aléatoire. L'analyse spectrale est donc simplifiée puisqu'elle ne nécessite pas d'estimer la densité spectrale de puissance du signal échantillonné. Pour plus d'informations concernant ce point, nous conseillons de vous référer aux travaux de thèse de Martin [Martin 1998], de Wojtiuk [Wojtiuk 2000] et de Nita [Nita 2000].

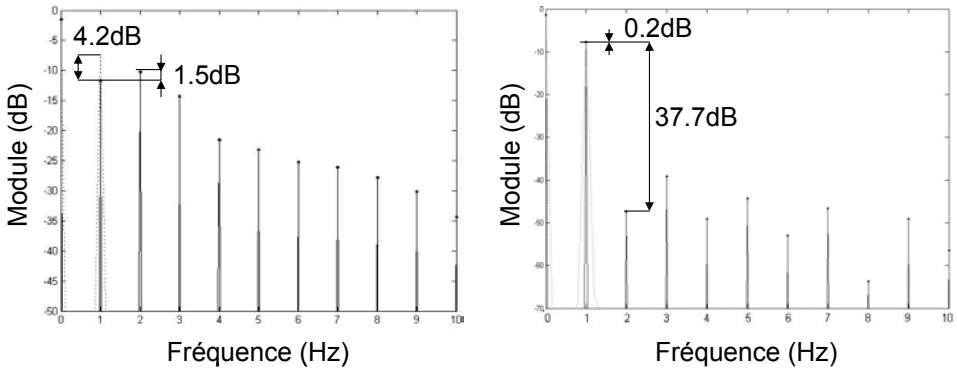


Figure 49 : Comparaison entre les spectres obtenus par GDFT (à gauche) et par TF du signal reconstruit à l'ordre 0 (à droite)

IV.2.2 Influence du quantum

La précision de la transformée de Fourier du signal reconstruit obtenue dans l'exemple précédent dépend du processus d'échantillonnage donc du signal d'entrée. S'il est possible par simulation d'estimer les performances de la reconstruction d'un signal donné, il n'est malheureusement pas possible de connaître les propriétés du signal d'entrée pour assurer un niveau de distorsion donnée. La précision de l'analyse et la qualité du signal reconstruit ne sont pas prédéfinissables. Pour illustrer ce propos nous avons étudié, pour un signal sinusoïdal simple, l'erreur relative sur l'amplitude du fondamental (Figure 50) ainsi que le taux de rejection du premier pic créé par la distorsion i.e. l'harmonique 2 (Figure 51). Le signal est échantillonné puis reconstruit par un bloqueur d'ordre 0.

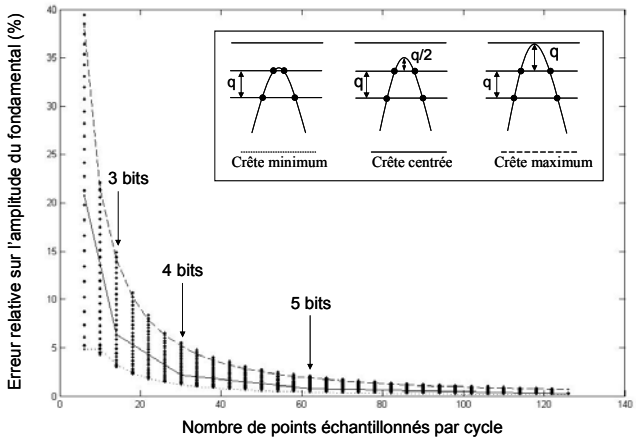


Figure 50 : Erreur relative sur l'amplitude du fondamental en fonction du nombre de points échantillonnés par cycle d'un signal sinusoïdal

Dans un premier temps, nous pouvons remarquer grâce à la représentation en fonction du nombre de points échantillonnés par cycle qu'il y a plusieurs erreurs possibles pour un même nombre de points. En effet, selon son amplitude, un signal croise un certain nombre de niveaux lors de l'échantillonnage. Or l'amplitude peut augmenter ou diminuer sans pour autant changer le nombre de niveaux croisés. Nous avons ainsi fait figurer sur les graphiques trois cas particuliers : quand la crête du signal dépasse à peine un niveau (trait en pointillé), quand la crête est symétrique c'est-à-dire entre deux niveaux (trait plein) et quand la crête tangente le niveau suivant sans le croiser (trait interrompu).

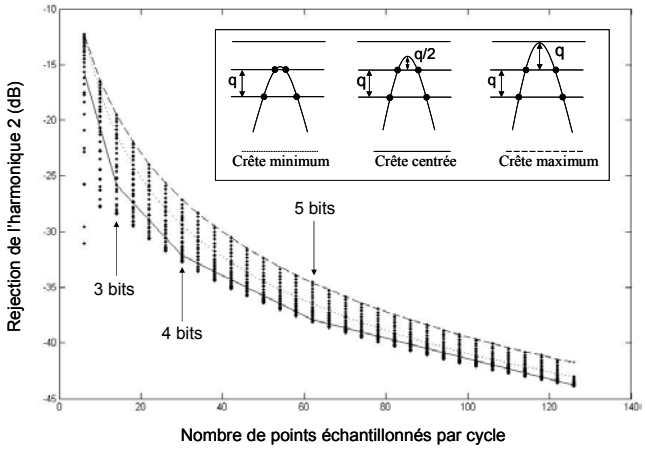


Figure 51 : Taux de rejection de l'harmonique 2 en fonction du nombre de points échantillonnés par cycle d'un signal sinusoïdal

Par ailleurs nous avons également représenté les convertisseurs 3 bits (7 niveaux), 4 bits (15 niveaux) et 5 bits (31 niveaux) utilisés dans leur pleine échelle. Ceci montre que la distorsion est indépendante d'une hypothèse temporelle sur les intervalles de temps puisqu'un nombre fixe de points cycle induit une fréquence moyenne d'échantillonnage constante ; d'autant plus qu'augmenter la fréquence moyenne d'échantillonnage ne fait pas forcément diminuer les erreurs. En effet, la distorsion dépend au contraire du rapport entre l'amplitude du signal et la valeur du quantum : plus il est grand, plus la distorsion est atténuée. Cependant, nous pouvons également observer que si les erreurs sont systématiquement bornées par celle correspond à un signal dont « la crête est maximum » (Tableau 3), il n'a pas une configuration unique (un rapport donné) minimisant tous les effets de la distorsion. Ceci prouve que faire varier le rapport entre l'amplitude et le quantum n'a pas une action homogène sur la distorsion. Et cela est d'autant plus vrai quand le signal contient d'autres composantes fréquentielles puisqu'il faut alors tenir compte des autres rapports entre l'amplitude de chaque harmonique et le quantum.

Résolution du convertisseur	**3 bits**	**4 bits**	**5 bits**
Erreur maximale sur le fondamental	15%	6%	2%
Taux de rejection minimum de h2	-19 dB	-28 dB	-34 dB

Tableau 3 : Récapitulatif des erreurs maximales en fonction de la résolution du convertisseur

IV.2.3 Equivalence avec un convertisseur classique

Pour généraliser l'étude de la distorsion à l'ordre 0, un élément de solution est de considérer l'erreur de reconstruction (comprise entre 0 et un quantum) équivalente au bruit généré par un convertisseur classique ayant le même nombre de niveaux de quantification *M* c'est-à-dire un convertisseur $log_2(M+1)$ bits. Nous supposons que l'incertitude sur les instants d'échantillonnage engendrée par la quantification du temps est négligeable dans la mesure où le bruit de quantification du temps est très inférieur (le circuit réalisé par exemple est équivalent à un convertisseur classique 10 bits) au bruit de quantification de l'amplitude (le circuit est un convertisseur 4 bits). La distorsion créée par la reconstruction peut alors être approchée en exprimant le rapport signal sur bruit selon la formule standard pour un signal sinusoïdal : $SRN(dB) = 6{,}02\ log_2(M+1) + 1{,}76$. Toutefois la différence fondamentale avec un convertisseur classique est que le bruit de quantification n'est plus un bruit blanc étalé sur tout le spectre. De plus, à la différence d'un convertisseur classique, les points de discontinuité du signal reconstruit i.e. aux instants d'échantillonnage, correspondent exactement en amplitude et en temps au signal d'origine (en négligeant encore une fois la quantification du temps). Ainsi, en lissant le signal reconstruit par un filtre passe bas dans les deux cas, les composantes haute fréquence seront supprimées. Mais la puissance du bruit de quantification restera inchangée dans la bande passante dans le cas de l'échantillonnage régulier, tandis que la distorsion sera atténuée dans le cas de l'échantillonnage par traversée de niveaux comme le montre la figure suivante.

Tsividis [Tsividis 2004] reprend ce principe pour montrer les avantages d'une nouvelle méthode de traitement numérique du signal basé sur le temps continu (en anglais *Continuous Time Digital Signal Processing*). Le principe est présenté en introduction du chapitre suivant concernant le filtrage numérique à réponse impulsionnelle finie.

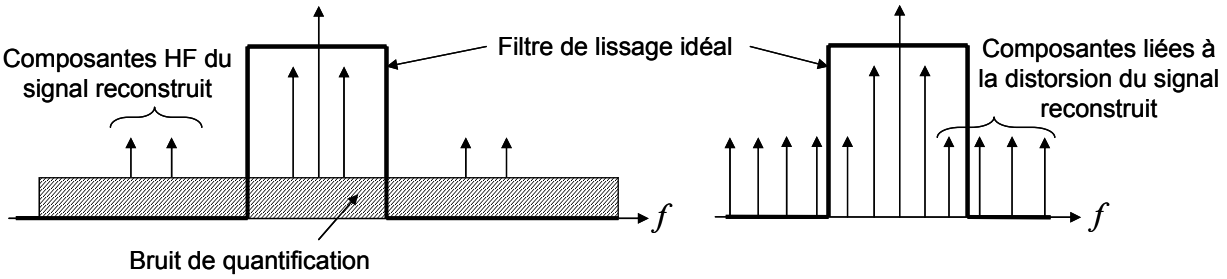


Figure 52 : Effet du lissage d'un signal reconstruit par bloqueur d'ordre 0 dans le cas régulier (à gauche) et dans le cas non uniforme par traversée de niveaux (à droite)

IV.3 Conclusion

Finalement, nous pouvons conclure que pour une valeur de quantum donnée, c'est-à-dire pour un nombre de niveaux fixé lors de la conversion analogique-numérique, il est possible de reconstruire en pratique un signal à temps continu avec une erreur donnée. Il important de noter que la reconstruction n'est plus un système linéaire et invariant dans le temps. Ceci implique qu'il n'est plus possible de représenter dans le domaine fréquentiel la reconstruction comme le filtrage du spectre du signal échantillonné car il y a autant de réponses impulsionnelles (donc de réponses en fréquence) qu'il y a d'échantillons. La notion de repliement de spectre n'a donc plus de sens dans l'échantillonnage par traversée de niveaux. En revanche, le point clé de la reconstruction qui était lié au choix de la fréquence d'échantillonnage est transposé au choix de la valeur du quantum. Dans le flot de conception du convertisseur analogique-numérique [Allier 2003], elle ne dépendait que du théorème de Beutler (lié à une reconstruction théorique précise); elle devra dorénavant prendre également en compte les spécificités du convertisseur numérique-analogique. Ceci montre que la conception des éléments d'une chaîne de traitement du signal échantillonnée non uniformément par traversée de niveaux doit être regroupée et non pas séparée. Enfin, la reconstruction par interpolation a montré qu'il était possible d'effectuer numériquement des traitements à temps continu à partir des échantillons (amplitude, intervalle de temps). Nous partirons donc de ce constat pour inclure l'information des intervalles de temps dans le traitement du signal échantillonné.

CHAPITRE V

Filtrage numérique à réponse impulsionnelle finie

Ce chapitre ainsi que le suivant sont dédiés au traitement des données échantillonnées non uniformément par traversée de niveaux. Nous considérons dans un premier temps le filtrage à réponse impulsionnelle finie. En effet, le filtrage est une des applications les plus élémentaires que l'on peut effectuer sur un signal – aussi bien à temps continu qu'à temps discret. Nous avons choisi d'étudier le filtrage à réponse impulsionnelle finie, car parmi les filtres numériques, ce sont les plus simples à réaliser : d'une part la sortie ne dépend que des échantillons d'entrée – il s'agit du produit de convolution entre l'entrée et la réponse impulsionnelle – et d'autre part, ils sont toujours stables.

La première partie est donc consacrée à la définition d'un produit de convolution à temps discret dédié aux signaux échantillonnés non uniformément. Celui-ci consiste à calculer à partir de signaux échantillonnés non uniformément, un produit de convolution analogique à un instant donné en interpolant les signaux à temps discret, en les multipliant puis en calculant leur intégrale. Afin de ne pas expliciter l'interpolation, nous introduisons un algorithme itératif qui décompose le résultat en une somme d'aires élémentaires obtenues directement à partir des échantillons. Nous étudions pour des interpolations simples les algorithmes, leurs complexités combinatoires ainsi que l'erreur maximale obtenue dans le pire cas.

Dans la seconde partie, nous définissons un filtre à réponse impulsionnelle finie basé sur le produit de convolution utilisant une interpolation d'ordre 0. Nous étudions ses caractéristiques (réponse impulsionnelle, réponse en fréquence) et en déduisons une architecture matérielle détaillée.

V.1 Introduction

Le filtrage est une opération de base en traitement du signal aussi bien sur des systèmes en temps continu qu'en temps discret. Dans le cadre de l'échantillonnage régulier, le filtrage est clairement défini [Oppenheim *et al.* 1995] depuis des années permettant de définir des relations entre les échantillons d'entrée et de sortie.

Nous nous sommes donc logiquement intéressé à cette opération dans le cadre spécifique de l'échantillonnage non uniforme par traversée de niveaux. Parmi les deux grandes catégories de filtres numériques – à réponse impulsionnelle finie RIF ou infinie RII, nous avons choisi dans un premier temps d'étudier les filtres RIF car ils possèdent la propriété fondamentale d'être stable inconditionnellement : la sortie courante ne dépendant pas des sorties précédentes, ne peut pas diverger comme le montre l'équation suivante :

$$y_n = \sum_{i=0}^{N-1} h_i x_{n-i} \qquad \text{Eq. (100)}$$

Par ailleurs il faut noter également que les filtres RIF sont les seuls dont la phase peut être linéaire, leur donnant un intérêt dans certaines applications audio ou vidéo comme l'égalisation.

Les travaux sur le filtrage numérique de type RIF dédiés aux signaux échantillonnés non uniformément sont rares. Nous citerons simplement ceux effectués par Tsividis [Tsividis 2004], [Li *et al.* 2005], [Tsividis *et al.* 2005]. Comme nous l'avons expliqué dans le chapitre précédent, il propose de travailler à temps continu en quantifiant, par un bloqueur d'ordre 0, le signal d'entrée au lieu de l'échantillonner. Il peut ainsi se libérer des contraintes liées au repliement de spectre et remplacer le bruit de quantification survenant lors de l'échantillonnage par une distorsion. Pour filtrer le signal d'entrée, il propose d'utiliser la formule de l'équation (79) habituellement destinée à la reconstruction en remplaçant respectivement les échantillons d'entrée par les coefficients de la réponse en fréquence (échantillonnée régulièrement à la fréquence $F_e = 1/T_e$) et la fonction de reconstruction par le signal d'entrée. L'équation (100) devient alors :

$$y(t) = \sum_{n=0}^{N-1} h_n x(t - nT_e) \qquad \text{Eq. (101)}$$

Cette formule n'est jamais utilisée en pratique car il n'est pas possible de réaliser un retard pur sur un signal quelconque dans le domaine analogique. En revanche dans le domaine numérique, il est connu qu'un retard peut être implémenté par une série d'inverseurs cascadés. Ainsi, en décomposant le signal d'entrée sur L niveaux en L signaux binaires à temps continu (*bit waveforms* en anglais) selon :

$$x(t) = \sum_{l=1}^{L} 2^{-l} b_l(t) \qquad \text{Eq. (102)}$$

il peut reformuler l'équation (101) :

$$y(t) = \sum_{l=1}^{L} 2^{-l} \left(\sum_{n=0}^{N-1} h_n(t) b_l(t - nT_e) \right) \qquad \text{Eq. (103)}$$

et en déduire une architecture de filtre à réponse impulsionnelle finie. En pratique, la décomposition du signal en signaux binaire repose sur un dispositif de conversion analogique-numérique proche du CANA. Cependant, sa méthode souffre de deux inconvénients majeurs. Tout d'abord, pour réaliser un retard, il faut cascader beaucoup d'inverseurs (typiquement le délai d'un inverseur est de 0,1ns mais peut varier en fonction de la technologie utilisée). Or pour l'ensemble du filtre, il doit en concevoir $(N-1)L$ identiques. En partant de son exemple de filtre à réponse impulsionnelle finie échantillonnée à $T_e = 125\mu s$, à 28 coefficients, il lui faudrait, pour 15 niveaux de quantification, 405 blocs « retard ». De plus chaque signal binaire retardé doit être multiplié à l'un des coefficients impliquant NL multiplieurs (dans son exemple, 420 multiplieurs), puis additionner entre eux par additionneur à NL entrées (i.e. à 420 entrées).

Par ailleurs, ne voulant concevoir ni un filtre numérique (signal d'entrée et réponse impulsionnelle à temps discret) ni un filtre analogique (signal d'entrée et réponse impulsionnelle à temps continu), il laisse le signal d'entrée à temps continu mais échantillonne la réponse impulsionnelle. Pour cette dernière, la théorie de l'échantillonnage uniforme s'applique donc, impliquant la périodisation de la réponse en fréquence. Ainsi n'importe quel filtre sera répliquer dans des bandes de fréquence autour des multiples de la fréquence d'échantillonnage où le signal d'entrée ne pourra pas être traité. Le seul moyen de se prémunir contre ces artéfacts de calcul est que la fréquence maximale du signal à traiter soit au pire la moitié de la fréquence d'échantillonnage i.e. respecter le théorème de Shannon.

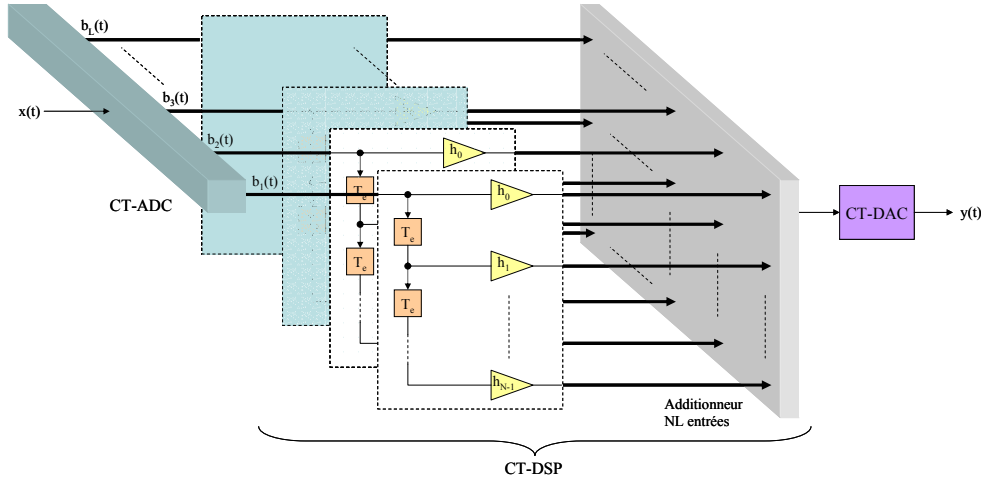


Figure 53 : Architecture de filtrage RIF à temps continu

Lorsque l'échantillonnage est non uniforme, il n'est plus possible d'utiliser une équation linéaire. Il faut donc élaborer une nouvelle formulation prenant en compte la spécificité de la non uniformité c'est-à-dire, pour le cas de l'échantillonnage par traversée de niveaux, de l'information des intervalles de temps. Puisque ni un filtre purement numérique, ni un filtre numérique à temps continu n'est utilisable, nous sommes donc revenus à la définition d'un filtre analogique en interpolant le signal d'échantillonné et la réponse impulsionnelle, pour traiter deux signaux à temps continu.

V.2 Produit de convolution asynchrone

V.2.1 Produit de convolution analogique

Dans le domaine analogique, le signal issu du produit de convolution de deux signaux est défini par la relation suivante :

$$y(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t-\tau) d\tau \qquad \text{Eq. (104)}$$

Théoriquement, le signal de sortie est calculé pour tous t et τ réels mais peut aussi l'être pour n'importe quel temps t appartenant à un ensemble discret $\{t_n\}$. De plus, si la réponse impulsionnelle est longueur finie T_h, le produit de convolution sera à durée limitée n'utilisant que des échantillons du signal d'entrée compris dans l'intervalle $[t_n; t_n\text{-}T_h]$. Nous utilisons cette propriété pour définir un produit de convolution à temps continu de signaux échantillonnés non uniformément.

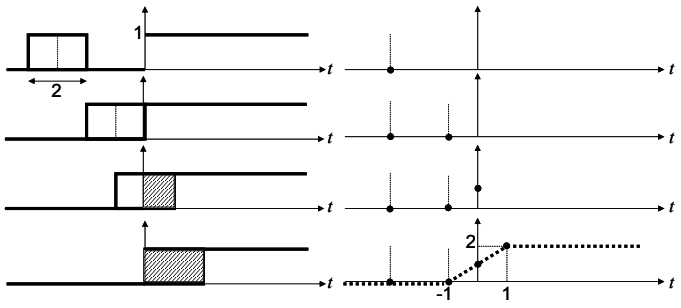


Figure 54 : Exemple du produit de convolution entre une fonction porte et l'échelon d'Heavyside

V.2.2 Produit de convolution asynchrone

V.2.2.1 Définition

Nous définissons un produit de convolution numérique pour des signaux échantillonnés non uniformément dans le temps à partir du produit de convolution analogique équivalent des signaux interpolés.

Soient *x* et *h*, deux fonctions d'entrée définies à temps continu par *x(t)* et *h(t)* et à temps discret par $\begin{cases} ax_n = x(tx_n) \\ dtx_n = tx_n - tx_{n-1} \end{cases}$ et $\begin{cases} ah_n = h(th_n) \\ dth_n = th_n - th_{n-1} \end{cases}$. La fonction de sortie *y* représentée en temps discret par $\begin{cases} ay_n = y(ty_n) \\ dty_n = ty_n - ty_{n-1} \end{cases}$ est alors définie comme le résultat du produit de convolution entre deux nouvelles fonctions interpolées à temps continu $\hat{x}(t)$ et $\hat{h}(t)$:

$$ay_n = y(ty_n) = \int_{-\infty}^{+\infty} \hat{x}(\tau) \cdot \hat{h}(ty_n - \tau) d\tau \qquad \text{Eq. (105)}$$

Il important de noter d'une part que l'instant de sortie correspond au décalage introduit dans l'intégrale, et d'autre part que pour calculer chaque nouvelle valeur de sortie, deux nouvelles interpolations sont à effectuer puisqu'elles dépendent de l'instant ty_n.

V.2.2.2 Condition de fin de calcul

L'équation (105) montre que pour calculer une sortie, il faut intégrer le produit des fonctions pendant une durée infinie. Or avec un système numérique, quel que soit le traitement considéré, un calcul est toujours à durée limitée, c'est pourquoi un produit de convolution n'est calculable que si l'un des signaux est de durée finie. Parmi les deux signaux disponibles, nous choisissons donc d'imposer par convention la réponse impulsionnelle finie à la fonction h : ainsi lorsque les N_h échantillons auront été utilisés, le calcul sera achevé. Nous appelons T_h la largeur temporelle de la fonction h.

V.2.2.3 Initialisation

L'instant d'échantillonnage de la sortie est un paramètre crucial dans le calcul car il correspond au décalage introduit dans le produit de convolution. Pour simplifier cette donnée à imposer au démarrage, nous choisissons de calculer une sortie pour chaque nouvel échantillon du signal x. Les instants d'échantillonnage de sortie sont donc égaux aux instants d'échantillonnage d'entrée. En outre, il n'y a pas d'opération à effectuer sur les intervalles de temps de sortie puisqu'ils sont aussi égaux aux intervalles de temps d'entrée :

$$\begin{cases} ty_n = tx_n \\ dty_n = dtx_n \end{cases} \qquad \text{Eq. (106)}$$

Au démarrage d'un calcul, le premier échantillon de la fonction h préalablement *retournée* sera décalé d'une durée tx_n, donc synchronisé à l'instant de l'échantillon en cours. La Figure 55 montre la position des échantillons des deux signaux pour le calcul de deux sorties successives.

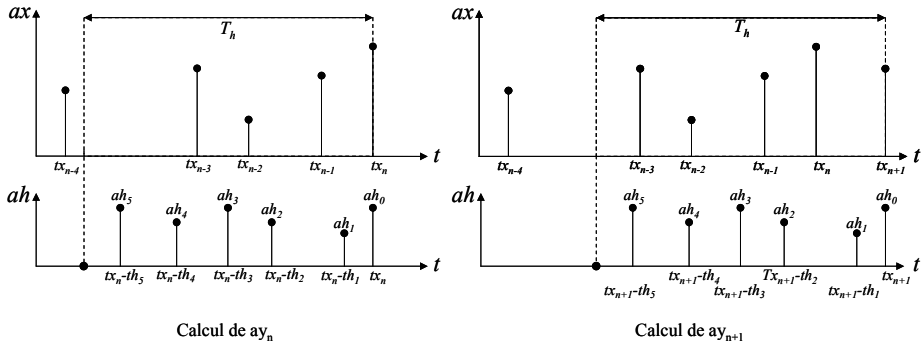


Figure 55 : Synchronisation des signaux à l'initialisation d'un calcul

V.2.2.4 Choix des interpolations

Le principe du produit de convolution asynchrone repose donc sur le choix d'une interpolation des signaux d'entrée. Il existe de multiples interpolations possibles. Toutefois, comme l'objectif principal de notre travail est la conception d'une chaîne de traitement du signal implémentable, nous nous limitons à des interpolations produisant des algorithmes de complexités combinatoires réduites :

- Interpolation d'ordre 0,
- Interpolation *au point le plus proche* que nous noterons 0.5,
- Interpolation linéaire d'ordre 1.

Pour des considérations pratiques évidentes, nous utiliserons la même interpolation pour les deux fonctions afin de ne pas multiplier les combinaisons possibles.

V.2.2.5 Interpolation des points manquants

Pour faciliter le calcul de la sortie, c'est-à-dire le calcul de l'aire du produit des deux fonctions interpolées, l'interpolation d'un signal doit dépendre à la fois du signal considéré et de l'autre signal. En effet, pour calculer le produit rapidement, il faut que les deux fonctions interpolées soient décomposables par morceaux de même taille. Ceci implique donc qu'un signal doit aussi être échantillonné aux instants de l'autre signal comme nous l'illustrons sur la Figure 56. Ainsi après interpolation, les deux fonctions $\hat{x}(t)$ et $\hat{h}(tx_n - t)$ sont, en temps discret, définies par les relations suivantes :

$$\begin{cases} \begin{cases} t\hat{x}_k = tx_{n-i} \\ a\hat{x}_k = \hat{x}(tx_{n-i}) = ax_{n-i} \end{cases} \\ \begin{cases} t\hat{x}_k = tx_n - th_j \\ a\hat{x}_k = \hat{x}(tx_n - th_j) \end{cases} \\ dt\hat{x}_k = t\hat{x}_k - t\hat{x}_{k-1} \end{cases} \text{ et } \begin{cases} \begin{cases} t\hat{h}_k = tx_{n-i} \\ a\hat{h}_k = \hat{h}(tx_n - tx_{n-i}) \end{cases} \\ \begin{cases} t\hat{h}_k = tx_n - th_j \\ a\hat{h}_k = \hat{h}(tx_n - th_j) = ah_j \end{cases} \\ dt\hat{h}_k = t\hat{h}_k - t\hat{h}_{k-1} \end{cases} \qquad \text{Eq. (107)}$$

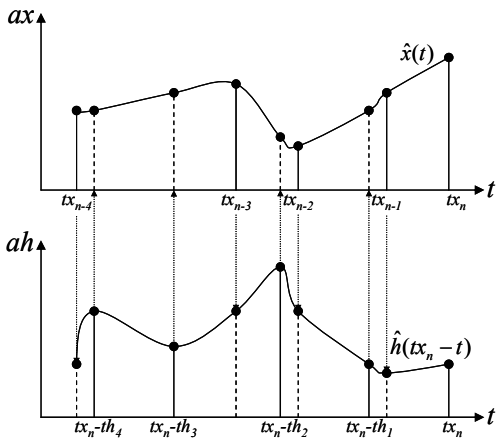


Figure 56 : Echantillonnage des points manquants par interpolation

Notons d'une part que les valeurs de $t\hat{x}_k$ et $t\hat{h}_k$ sont égales car les deux signaux sont cette fois-ci échantillonnés aux mêmes instants et d'autre part qu'il y a deux nouvelles interpolations à chaque nouveau calcul du fait de la resynchronisation de la réponse impulsionnelle.

V.2.2.6 Calcul d'une intégrale numérique

Conformément à l'équation (105), l'amplitude du signal de sortie est égale à la somme des surfaces du produit des fonctions dans chaque morceau élémentaire. Deux approches peuvent alors être considérées pour déterminer la surface d'une plage :

- en partant du produit des échantillons pour calculer une surface simplifiée,
- en partant du produit des fonctions interpolées pour calculer une surface complète.

V.2.2.6.1 Intégrale simplifiée

Cette approche consiste à multiplier terme à terme les signaux ré-échantillonnés décrits par les relations de l'équation (107) afin d'obtenir directement l'image échantillonnée du produit des fonctions interpolées. La valeur de la sortie est ensuite déterminée simplement en sommant des aires élémentaires. Le calcul est alors équivalent à l'intégration d'un signal numérique. Nous considérerons, dans cette perspective, deux méthodes :

- l'utilisation de rectangles élémentaires,
- l'utilisation de trapèzes élémentaires.

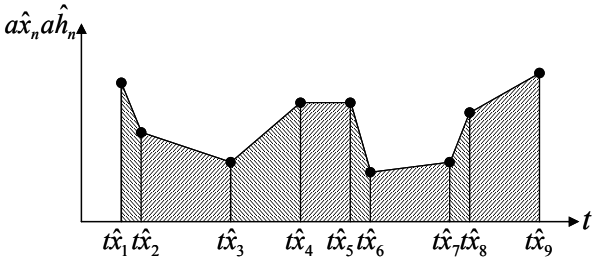


Figure 57 : Calcul simplifié de l'intégrale par multiplication des échantillons et utilisation de trapèzes élémentaires

V.2.2.6.2 Intégrale complète

La seconde approche consiste à multiplier plage par plage les deux fonctions interpolées en temps continu. La fonction ainsi obtenue est ensuite intégrée pour calculer la valeur de l'échantillon de sortie. Bien qu'il s'agisse d'une intégrale à temps continu, connaissant l'équation des fonctions interpolées, nous pouvons déterminer facilement la primitive générique du produit puis en déduire la valeur de la surface en fonction des amplitudes et intervalles de temps des deux signaux ré-échantillonnés.

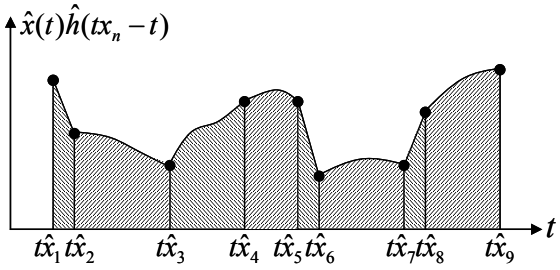


Figure 58 : Calcul complet de l'intégrale par multiplication des fonctions interpolées

V.2.2.7 Algorithme itératif

D'après ce que nous venons de présenter, le calcul d'un échantillon de sortie doit suivre le protocole suivant [Aeschlimann 2002] :

1. Interpolation du signal *x* aux instants du signal *h* retourné et décalé de tx_n.
2. Interpolation du signal *h* retourné et décalé aux instants du signal *x*
3. Fusion des deux séries d'instants tx_{n-i} et $(tx_n - th_j)$ pour déterminer les instants $t\hat{x}_k$
4. Somme des surfaces des plages élémentaires définies par les nouveaux intervalles $\{t\hat{x}_k - t\hat{x}_{k-1}\}$

Les étapes 1 à 3 requièrent parallèlement le balayage des deux signaux pendant une durée T_h. Ensuite pendant l'étape 4, le même balayage est exécuté pour calculer la somme de toutes les surfaces. Nous proposons donc plutôt de procéder au calcul direct d'une surface, en déterminant coup par coup les plages élémentaires. En effet, lorsque deux échantillons des signaux sont synchronisés (par exemple au départ), il suffit simplement de comparer les deux intervalles de temps pour déterminer la plage élémentaire correspondant au plus petit des deux. La surface élémentaire est alors calculée à partir des données en cours qui sont mises à jour en vue de la plage suivante : l'échantillon du signal correspondant à l'intervalle le plus petit, n'est plus valide; son indice est donc incrémenté pour passer à l'échantillon suivant. En revanche l'échantillon de l'autre signal est encore valide; ses grandeurs sont juste modifiées : son amplitude est interpolée suivant une des techniques tandis que son intervalle de temps est réduit d'une durée égale à la plage en cours pour le resynchroniser avec l'échantillon nouvellement arrivé. Si les deux intervalles de temps sont égaux, aucune donnée n'est mise à jour, deux nouveaux échantillons sont introduits pour le prochain calcul.

Finalement nous en avons extrait un algorithme itératif présenté par un organigramme Figure 59 [Aeschlimann *et al.* 2004] : lorsqu'un nouvel échantillon en entrée est prélevé, un nouveau calcul doit commencer. Pour s'assurer que lors de la première comparaison, les deux intervalles de temps sont synchronisés, les indices i et j sont mis à zéro. Parallèlement, les données du nouvel échantillon sont stockées en mémoire. Chaque itération débute par la recherche du minimum des intervalles de temps en cours. Puis en fonction de cette valeur, l'aire de la plage est calculée et accumulée aux surfaces des itérations précédentes. Lorsque tous les échantillons de la fonction h ont été utilisés, la boucle s'achève : l'amplitude de sortie est égale à valeur des surfaces accumulées.

Cet algorithme offre deux avantages :

1. Il balaye une seule fois les deux signaux.
2. Il n'utilise que les données des signaux x et h : il ne nécessite pas en effet le calcul des instants $t\hat{x}_k$ pour déterminer les plages élémentaires.

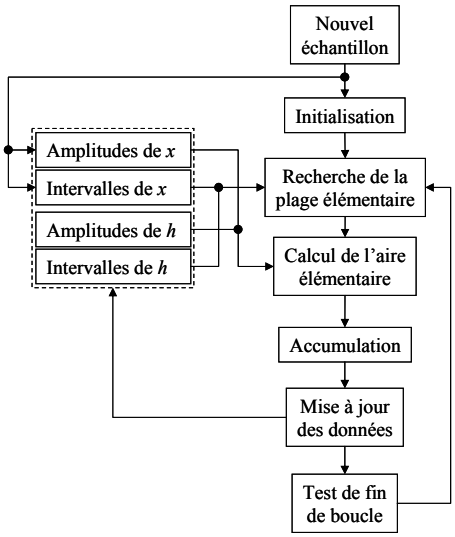


Figure 59 : Organigramme générique de l'algorithme itératif

V.2.2.8 Complexité combinatoire

Pour déterminer l'influence de cet algorithme sur l'activité du système, il faut étudier sa complexité combinatoire, c'est-à-dire le nombre d'opérations de base (addition, multiplication, décalage…) nécessaire au calcul d'un échantillon de sortie. Bien qu'en termes de conception, un multiplieur et un additionneur n'aient pas le même coût (surface, consommation…), nous ferons cette hypothèse. Par ailleurs, nous ne compterons ni les comparaisons ni les compteurs.

D'autre part, même si le nombre d'opérateurs dépend des interpolations utilisées, il est possible, a priori, de calculer le nombre d'itérations en déterminant le nombre de plages élémentaires. En effet, à la fin du calcul d'une sortie, nous savons qu'il y aura eu N_h échantillons utilisés de la fonction h et N_x échantillons de la fonction x sur une fenêtre temporelle $[tx_n\text{-}T_h\ ;\ tx_n]$. Par ailleurs, comme pour chaque nouvelle sortie, il y a de nouvelles interpolations, N_x n'est pas constant. Le nombre d'itérations varie donc au cours du temps :

$$N_x\big|_{[tx_n-T_h;tx_n]} = \max_k\left\{\sum_{i=0}^{k} dtx_{n-i} \le T_h\right\} \qquad \text{Eq. (108)}$$

Toutefois, nous pouvons estimer le nombre d'itérations B_n effectuées pour le calcul de la sortie à l'instant tx_n :

$$B_n = N_h + N_x\big|_{[tx_n-T_h;tx_n]} - C_n \qquad \text{Eq. (109)}$$

Le nombre B_n correspond en effet à la somme des échantillons des deux fonctions prélevés à des instants différents. Il est obtenu en ajoutant le nombre total d'échantillons de chaque fonction, puis en retranchant le nombre de fois C_n où deux échantillons sont pris au même instant sur l'intervalle $[tx_n - T_h ; tx_n]$. De plus, étant donné que statistiquement il y a une probabilité très faible que deux échantillons soient disposés au même instant, C_n tend vers 1. Le nombre B_n peut alors être approché par la relation suivante :

$$B_n \approx N_h + N_x\big|_{[tx_n - T_h ; tx_n]} - 1 \qquad \text{Eq. (110)}$$

Le nombre d'échantillons d'entrée N_x, utilisé pour calculer une sortie, dépend de l'instant de synchronisation tx_n, pour une valeur de T_h donnée. Nous proposons donc d'estimer N_x en fonction de la valeur moyenne des intervalles de temps de la fonction x sur la fenêtre $[tx_n - T_h ; tx_n]$:

$$N_x\big|_{[tx_n - T_h ; tx_n]} \approx \frac{T_h}{moy(dtx\big|_{[tx_n - T_h ; tx_n]})} \qquad \text{Eq. (111)}$$

Or, en considérant une fenêtre temporelle de largeur T_h, le nombre d'échantillons N_h dépend également des intervalles de temps de la fonction h :

$$N_h = \frac{T_h}{moy(dth)} \qquad \text{Eq. (112)}$$

Ainsi, en regroupant les relations (111) et (112) dans l'équation (110), nous pouvons estimer le nombre d'itérations effectuées par l'algorithme en fonction des intervalles de temps des deux fonctions et du nombre de points de la fonction à durée limitée permettant la condition de fin de boucle :

$$B_n \approx N_h \left(1 + \frac{moy(dth)}{moy(dtx\big|_{[tx_n - T_h ; tx_n]})} \right) - 1 \qquad \text{Eq. (113)}$$

Pour connaître le nombre total d'opérations O_n effectuées lors du calcul du $n^{ième}$ échantillon de sortie, il suffit de multiplier le nombre d'opérateurs N_{op} de la méthode choisie par le nombre d'itérations B_n :

$$O_n = N_{op} \cdot B_n \qquad \text{Eq. (114)}$$

V.2.2.9 Erreur de calcul

L'amplitude de sortie calculée par notre algorithme est une approximation du produit de convolution analogique. Les interpolations introduisent des erreurs à chaque itération entre l'intégrale exacte et la surface approchée. Nous proposons donc d'étudier l'erreur pendant une itération k entre le produit théorique et celui réalisé.

Pour déterminer l'erreur de calcul maximale, il faut étudier le cas pour lequel l'intégration donne le résultat le plus éloigné de celui obtenu. Nous posons les hypothèses suivantes afin d'éviter les cas irréalistes :

- Dans une plage donnée, les signaux sont toujours compris entre les valeurs des échantillons bornant la plage. Ils varient donc respectivement dans des intervalles Δ_x et Δ_h présumés constants.
- L'amplitude des échantillons suivants est, pour les deux fonctions, inférieure à celle des échantillons courants (les fonctions sont considérées décroissantes) afin d'éviter les compensations entre les amplitudes.
- L'intervalle de temps minimum est attribué arbitrairement au signal x.
- Nous noterons α_k le rapport de l'intervalle de temps de x sur l'intervalle de temps de h à l'itération k. D'une manière générale, α_k correspond au rapport entre l'intervalle de temps plus petit et l'intervalle le plus grand.
- Certains points utilisés pour le calcul de l'aire sont supposés être interpolés. Dans la mesure où nous tenons déjà compte de l'interpolation dans le choix du pire cas, nous considérons que les points sont exacts.
- L'erreur sera définie comme le rapport de la valeur absolue de la différence des aires sur l'aire du pire cas :

$$\varepsilon_M = \frac{\left|A - A_{pc}\right|}{A_{pc}} \qquad \text{Eq. (115)}$$

L'erreur maximale d'une technique donnée ne peut pas être mesurée quantitativement car elle dépend des paramètres des deux signaux (amplitude, intervalle de temps, écart d'amplitude, rapport cyclique). En revanche, à itération et signaux donnés, nous pourrons comparer les interpolations en normalisant l'erreur par rapport à celle de l'une des techniques.

V.2.2.10 Produit de convolution asynchrone d'ordre 0

Le produit de convolution asynchrone d'ordre 0 est basé sur l'utilisation d'une approximation de l'intégrale par un rectangle élémentaire et d'une mise à jour des amplitudes par une interpolation d'ordre 0. Comme la fonction interpolée dans une plage élémentaire est une constante, l'intégrale simplifiée est identique à l'intégrale complète.

V.2.2.10.1 Algorithme

Comme nous l'avons vu précédemment, une itération procède au calcul partiel d'une sortie en trois étapes successives (un organigramme détaillé est présenté Figure 60). La première étape consiste à rechercher l'intervalle de temps minimum entre les deux échantillons courants pour déterminer la plage élémentaire. Puis l'aire d'un rectangle dont la hauteur est le produit des échantillons et dont la largeur est la durée de la plage, est ajoutée à la valeur de la sortie calculée pendant l'itération précédente. Enfin les données sont mises à jour : l'indice de l'échantillon correspondant à l'intervalle de temps minimum est incrémenté pour passer au point suivant. Parallèlement, l'autre point voit son intervalle de temps réduit pour resynchroniser les échantillons tandis que son amplitude est maintenue constante (interpolation d'ordre 0).

Nous pouvons ainsi écrire le calcul de la sortie du produit de convolution asynchrone d'ordre 0 par la relation suivante :

$$\begin{cases} dty_n = dtx_n \\ ay_n = ay(tx_n) = \sum_{i,j}^{j=N_h} \min(dtx_{n-i}, dth_j) ax_{n-i} ah_j \end{cases} \qquad \text{Eq. (116)}$$

V.2.2.10.2 Complexité combinatoire

A l'ordre 0, nous pouvons facilement observer que deux multiplications et une addition sont nécessaires au calcul de l'aire et une addition pour mettre à jour les intervalles de temps. Ainsi au final, il faut quatre opérations par itération pour calculer une sortie. Comme nous savons qu'il y a B_n itérations, nous pouvons déduire que la complexité combinatoire du produit convolution à l'ordre 0 est :

$$O_{0,n} = 4B_n \qquad \text{Eq. (117)}$$

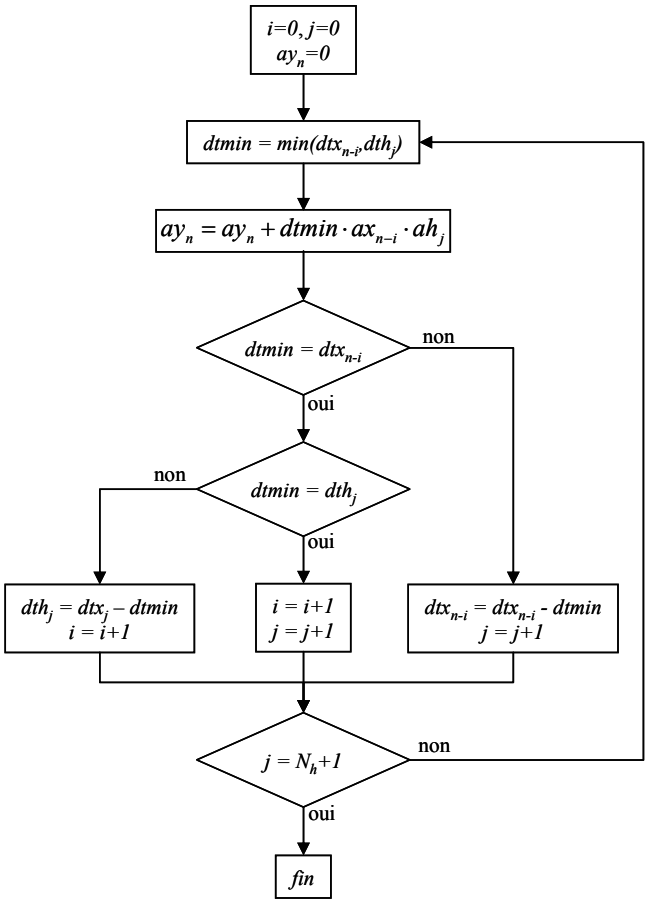


Figure 60 : Organigramme du produit de convolution asynchrone d'ordre 0

V.2.2.10.3 Erreur de calcul

A l'aide d'un dessin (Figure 61), nous pouvons remarquer que le pire cas correspond au calcul de l'aire du produit de deux fonctions constantes d'amplitudes réduite respectivement de Δ_x et Δ_h.

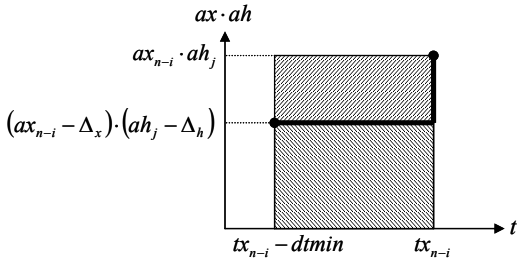


Figure 61 : Etude du pire cas de l'ordre 0

L'aire calculée par l'algorithme est définie par :

$$A_0 = dtmin \cdot ax_{n-i} \cdot ah_j \qquad \text{Eq. (118)}$$

Parallèlement, l'aire du pire cas est donnée par :

$$A_{pc} = dtmin \cdot \left(ax_{n-i} - \Delta_x\right) \cdot \left(ah_j - \Delta_h\right) \qquad \text{Eq. (119)}$$

Nous pouvons donc déduire des deux relations précédentes que l'erreur de calcul relative maximale du produit de convolution asynchrone d'ordre 0 vaut :

$$\varepsilon_{M0} = \frac{\Delta_h\left(ax_{n-i} - \frac{\Delta_x}{2}\right) + \Delta_x\left(ah_j - \frac{\Delta_h}{2}\right)}{\left(ax_{n-i} - \Delta_x\right)\left(ah_j - \Delta_h\right)} \qquad \text{Eq. (120)}$$

V.2.2.11 Produit de convolution asynchrone d'ordre 0.5

Le produit de convolution asynchrone d'ordre 0.5 utilise une interpolation *au point le plus proche* pour ré-échantillonner les points manquants. A la différence de la méthode précédente, il existe deux solutions pour calculer l'aire d'une plage élémentaire et donc deux algorithmes que nous allons décrire dans les sections suivantes. En effet, nous pouvons approcher l'aire exacte soit par l'aire du trapèze du produit des échantillons, soit par l'aire du produit des deux fonctions interpolées. Nous appellerons respectivement la première solution : *produit de convolution asynchrone d'ordre 0.5s* pour *simplifié* et la seconde : *produit de convolution asynchrone d'ordre 0.5c* pour *complet*.

V.2.2.11.1 Algorithme simplifié

Avec le calcul simplifié de l'aire, l'amplitude du signal de sortie correspond à la somme de surfaces de trapèzes. En fonction du rapport α_k entre l'intervalle de temps le plus petit et l'intervalle le plus grand, deux configurations se dessinent :

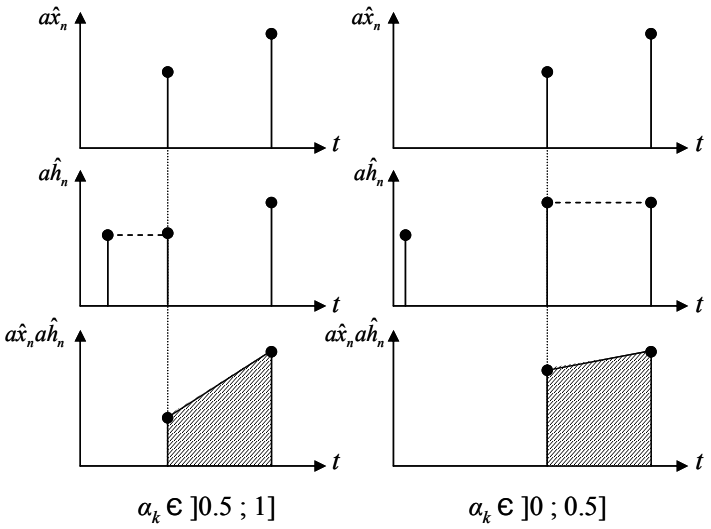


Figure 62 : Produit de convolution asynchrone d'ordre 0.5s

Il y a deux cas possibles pour calculer l'aire comme le montre la Figure 62.

1. Si le rapport entre les intervalles de temps est supérieur à 0.5 (partie gauche de la Figure 62) : les échantillons courants et suivants sont multipliés terme à terme, puis l'aire est extraite suivant la règle du trapèze.
2. Si le rapport est inférieur ou égal à 0.5 (partie droite de la Figure 62) : d'un coté, les échantillons courants des deux signaux sont multipliés entre eux tandis que de l'autre, l'échantillon suivant du signal le plus court est multiplié par l'échantillon courant du signal le plus long.

Il faut donc d'une part connaître le signal le plus court et d'autre part déterminer, en fonction de α_k, la configuration dans laquelle nous nous trouvons pour utiliser les bons échantillons dans le calcul de l'aire. Enfin la mise à jour des données est à effectuer en fonction du signal le plus court et du rapport α_k.

Comme la sélection des échantillons nécessaires au calcul et la mise à jour des données utilisent le même processus (mêmes comparaisons entre les intervalles de temps), nous proposons de passer une seule fois cette étape, de stocker les données (amplitudes suivantes, intervalles de temps suivants, incréments des indices) dans six variables locales, utiliser les données courantes et les variables locales pour calculer l'aire du trapèze et enfin mettre à jour les données courantes avec les variables locales. Nous obtenons ainsi l'organigramme suivant :

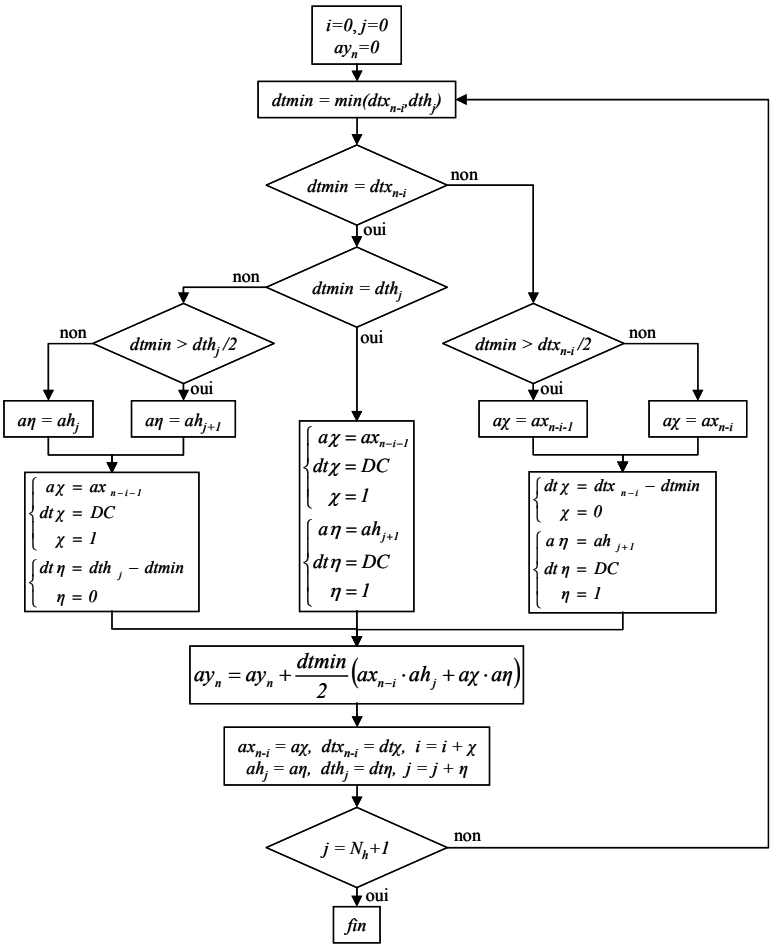


Figure 63 : Organigramme du produit de convolution asynchrone d'ordre 0.5s

Les variables locales contiennent respectivement pour les signaux *x* et *h* :

- les amplitudes nécessaires au calcul de l'aire et à la mise à jour des amplitudes courantes : $a\chi$ et $a\eta$,
- la valeur des intervalles de temps à mettre à jour : $dt\chi$ et $dt\eta$,
- la valeur de l'incrément des indices des signaux : χ et η.

Notons que si la valeur d'un incrément est égale à un, la mise à jour du point courant ne sert à rien puisqu'à l'itération suivante, un nouveau point sera utilisé. Ceci explique l'affection d'un *DC* (*Don't Care*) à certains intervalles de temps.

V.2.2.11.2 Algorithme complet

Avec le calcul complet de l'aire, l'amplitude du signal de sortie correspond à la somme des surfaces du produit des fonctions interpolées, c'est-à-dire à la somme des surfaces de rectangles. Bien que cette méthode ait l'air plus simple que la précédente, il se trouve que le nombre de

rectangles dans une plage élémentaire varie en fonction du rapport α_k. Il faut alors distinguer trois cas qui sont représentés sur la figure suivante :

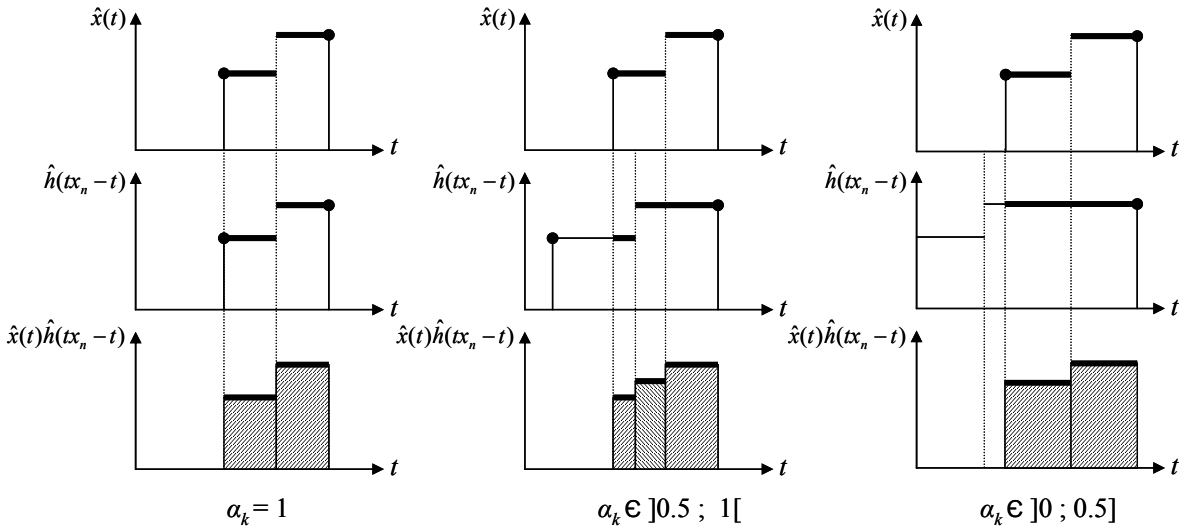


Figure 64 : Produit de convolution asynchrone d'ordre 0.5c

La première configuration (partie gauche de la Figure 64) correspond au cas où le rapport est égal à un, c'est-à-dire lorsque les deux intervalles de temps ont la même valeur ; il y a alors deux rectangles de même largeur. Nous pouvons remarquer que la surface du produit des fonctions est équivalente à celle calculée avec la méthode du trapèze lorsque le rapport était compris entre 0.5 et 1.

La seconde configuration (partie centrale de la Figure 64) correspond au cas où le rapport est compris entre 0.5 et 1. Il y a alors trois rectangles de largeurs respectives *dtmin*, $dtmin\left(\frac{1-\alpha_k}{2\alpha_k}\right)$ et $dtmin\left(\frac{2\alpha_k - 1}{2\alpha_k}\right)$.

Enfin la troisième configuration (partie droite de la Figure 64) correspond au cas où le rapport est inférieur ou égal à 0.5. Il y a alors deux rectangles de largeur égale. Nous pouvons encore remarquer que la surface est équivalente à celle calculée avec la méthode du trapèze lorsque α_k avait la même valeur.

La sélection des échantillons nécessaires au calcul et la mise à jour des données utilisent encore le même processus (mêmes comparaisons entre les intervalles de temps). Nous proposons alors de procéder à une seule analyse en stockant de nouveau les données (amplitude suivante, intervalle de temps suivant, incréments des indices) dans neuf variables locales (les six précédentes et trois nouvelles). Les données courantes et les trois nouvelles variables locales sont ensuite utilisées pour calculer l'aire des rectangles : en effet, l'aire totale A d'une plage est donnée (sous conditions) par la formule suivante :

$$\begin{aligned} A &= \frac{dtmin}{2}\left(ax_{n-i}\cdot ah_j\right)+dtmin\left(\frac{1-\alpha_k}{2\alpha_k}\right)\left(a1\cdot a2\right)+dtmin\left(\frac{2\alpha_k-1}{2\alpha_k}\right)\left(a1\cdot a3\right) \\ &= \frac{dtmin}{2}\left(ax_{n-i}\cdot ah_j+\frac{a1}{\alpha_k}\left(\left(1-\alpha_k\right)a2+\left(2\alpha_k-1\right)a3\right)\right) \end{aligned} \qquad \text{Eq. (121)}$$

où *a1*, *a2* et *a3* sont les trois variables locales représentant respectivement l'amplitude suivante du signal correspondant au *dtmin* et les amplitudes courante et suivante du signal ne correspondant pas au *dtmin*. Les six variables restantes servent à mettre à jour les données courantes. Finalement, nous obtenons l'organigramme suivant :

V.2.2.11.3 Complexités combinatoires

La complexité combinatoire dépend de la technique utilisée pour calculer l'aire.

Avec la méthode simplifiée, il faut trois multiplications et deux additions pour calculer la surface et une addition pour mettre à jour les intervalles de temps. Compte tenu des B_n itérations nécessaires au calcul complet de la sortie, nous pouvons en déduire que la complexité combinatoire du produit de convolution asynchrone d'ordre 0.5s est :

$$O_{0.5s,n} = 6B_n \qquad \text{Eq. (122)}$$

Avec la méthode complète, le nombre d'opérations nécessaires au calcul de la surface est augmenté : il faut en effet sept multiplications et cinq additions pour calculer l'aire d'une plage. En revanche, il faut toujours une seule addition pour mettre à jour les intervalles de temps. Au total, nous pouvons exprimer la complexité combinatoire du produit de convolution asynchrone d'ordre 0.5c :

$$O_{0.5c,n} = 13B_n \qquad \text{Eq. (123)}$$

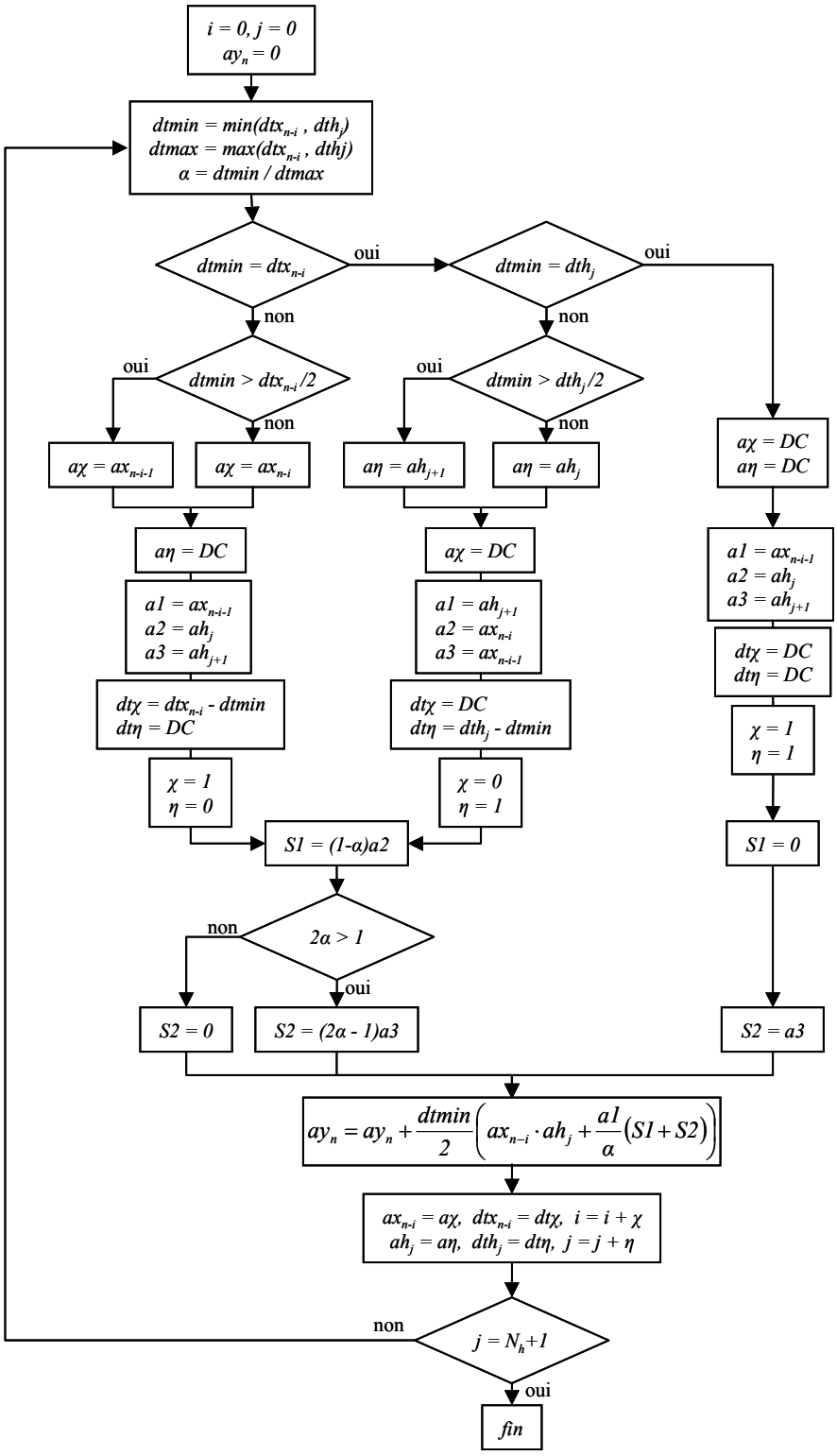


Figure 65 : Organigramme du produit de convolution asynchrone d'ordre 0.5c

V.2.2.11.4 Erreurs de calcul

Pour estimer les erreurs de calcul introduites par l'utilisation d'un produit de convolution asynchrone d'ordre 0.5, il faut distinguer, pour les deux méthodes, toutes les configurations en fonction du paramètre α_k.

Avec la méthode simplifiée, il y a deux configurations que nous illustrons sur la Figure 66, dans lesquelles l'aire au pire cas correspond à celle définie précédemment à l'ordre 0 (équation (119)). Il y a donc deux aires possibles :

$$A_{0.5s,1} = \frac{dtmin}{2} \cdot \left(ax_{n-i} \cdot ah_j + \left(ax_{n-i} - \Delta_x \right) \cdot \left(ah_j - \Delta_h \right) \right) \text{ si } \alpha_k > 0.5 \qquad \text{Eq. (124)}$$

$$A_{0.5s,2} = \frac{dtmin}{2} \cdot ah_j \cdot \left(2ax_{n-i} - \Delta_x \right) \text{ si } \alpha_k \leq 0.5 \qquad \text{Eq. (125)}$$

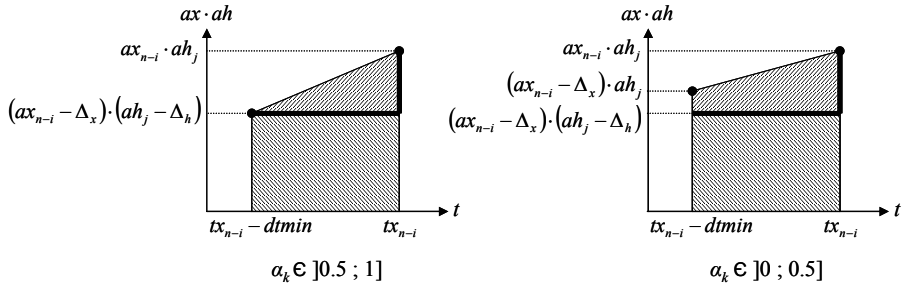


Figure 66 : Etude du pire cas de l'ordre 0.5s

Par conséquent, il faut considérer deux erreurs maximales possibles pour le produit de convolution asynchrone d'ordre 0.5s :

$$\varepsilon_{M0.5s,1} = \frac{1}{2} \cdot \left(\frac{ax_{n-i}}{ax_{n-i} - \Delta_x} \cdot \frac{ah_j}{ah_j - \Delta_h} - 1 \right) = \frac{\varepsilon_{M0}}{2} \text{ si } \alpha_k > 0.5 \qquad \text{Eq. (126)}$$

$$\varepsilon_{M0.5s,2} = \frac{1}{2} \cdot \left(\frac{ax_{n-i}}{ax_{n-i} - \Delta_x} \cdot \frac{ah_j}{ah_j - \Delta_h} + \frac{\Delta_h}{ah_j - \Delta_h} - 1 \right) = \frac{3\varepsilon_{M0}}{4} \text{ si } \alpha_k \leq 0.5 \qquad \text{Eq. (127)}$$

Avec la méthode complète, nous pouvons remarquer que l'aire calculée avec $\alpha_k = 1$ est identique à la première configuration ($\alpha_k > 0.5$) de la méthode simplifiée. Ce cas ne sera donc pas réétudié; nous poserons simplement :

$$A_{0.5c,1} = A_{0.5s,1} \qquad \text{Eq. (128)}$$

Il y a deux configurations distinctes que nous représentons sur la Figure 84, dans lesquelles l'aire au pire cas correspond encore une fois à celle définie précédemment à l'ordre 0 (équation (119)).

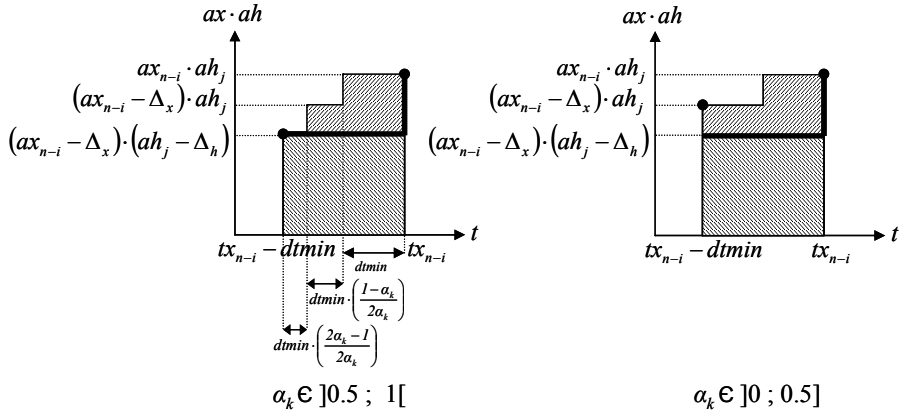


Figure 67 : Etude du pire cas de l'ordre 0.5c

Pour la première configuration, c'est-à-dire lorsque α_k appartient à l'intervalle *]0.5; 1[*, l'aire calculée est définie par l'équation suivante :

$$A_{0.5c,2} = \frac{dtmin}{2} \cdot \left(ax_{n-i} \cdot ah_j + (ax_{n-i} - \Delta_x) ah_j - \Delta_h \left(\frac{2\alpha_k - 1}{\alpha_k} \right) (ax_{n-i} - \Delta_x) \right) \qquad \text{Eq. (129)}$$

En revanche, lorsque α_k appartient à l'intervalle *]0; 0.5]*, l'aire calculée est égale à celle déjà définie dans le cas simplifié pour des valeurs identiques de α_k :

$$A_{0.5c,3} = A_{0.5s,2} \qquad \text{Eq. (130)}$$

Nous pouvons alors en déduire que les erreurs maximales introduites par le produit de convolution asynchrone d'ordre 0.5c sont :

$$\varepsilon_{M0.5c,1} = \varepsilon_{M0.5s,1} = \frac{\varepsilon_{M0}}{2} \text{ si } \alpha_k = 1 \qquad \text{Eq. (131)}$$

$$\varepsilon_{M0.5c,2} = \frac{1}{2} \cdot \left(\frac{ax_{n-i}}{ax_{n-i} - \Delta_x} \cdot \frac{ah_j}{ah_j - \Delta_h} - 1 + \frac{(1-\alpha_k)}{\alpha_k} \frac{\Delta_h}{ah_j - \Delta_h} \right) \text{ si } \alpha_k \in]0.5;\ 1[\quad \text{Eq. (132)}$$

$$\varepsilon_{M0.5c,3} = \varepsilon_{M0.5s,2} = \frac{3\varepsilon_{M0}}{4} \text{ si } \alpha_k \in]0;\ 0.5] \quad \text{Eq. (133)}$$

Nous pouvons conclure que l'erreur maximale du produit de convolution asynchrone d'ordre 0.5c est toujours bornée par :

$$\frac{\varepsilon_{M0}}{2} \leq \varepsilon_{M0.5c} \leq \frac{3\varepsilon_{M0}}{4} \quad \text{Eq. (134)}$$

V.2.2.12 Produit de convolution asynchrone d'ordre 1

Le produit de convolution asynchrone d'ordre 1 utilise une interpolation linéaire pour ré-échantillonner les points manquants. Comme pour l'ordre 0.5, il existe deux solutions pour calculer l'aire d'une plage élémentaire et donc deux nouveaux algorithmes. En effet, nous pouvons approcher l'aire exacte soit par l'aire du trapèze du produit des échantillons, soit par l'aire du produit des deux fonctions interpolées. Nous appellerons respectivement la première solution : *produit de convolution asynchrone d'ordre 1s* pour *simplifié* et la seconde : *produit de convolution asynchrone d'ordre 1c* pour *complet*.

V.2.2.12.1 Algorithme simplifié

Avec le calcul simplifié de l'aire, l'amplitude du signal de sortie correspond à la somme de surfaces de trapèzes. Nous en représentons le principe sur la Figure 68.

L'algorithme simplifié qu'il soit utilisé pour le produit de convolution asynchrone d'ordre 0.5 ou d'ordre 1, est fondé sur le principe qu'un des deux signaux est ré-échantillonné. Le nouveau point sert d'une part à calculer la surface de la plage courante et d'autre part à mettre à jour le signal en vue de l'itération suivante. Nous proposons alors de procéder, comme à l'ordre 0.5s, à une seule analyse en stockant de nouveau les données (amplitude suivante, intervalle de temps suivant, incréments des indices) dans six variables locales. De plus, comme l'interpolation d'ordre 1 nécessite la connaissance des pentes des signaux, il faut déterminer pour les deux signaux la différence entre les amplitudes du point courant et du point suivant. Pour le signal d'entrée, si nous considérons qu'il est échantillonné par traversée de niveaux, la différence vaut au signe près

un quantum noté q. En revanche, le second signal est censé représenter la réponse impulsionnelle ; il est donc défini en amont du traitement. Nous proposons simplement de stocker en plus de ses amplitudes et de ses intervalles de temps, la différence de ses amplitudes notées ah'_j. Nous obtenons finalement le même algorithme qu'à l'ordre 0.5s à l'interpolation près. L'organigramme de l'algorithme est présenté sur la Figure 69.

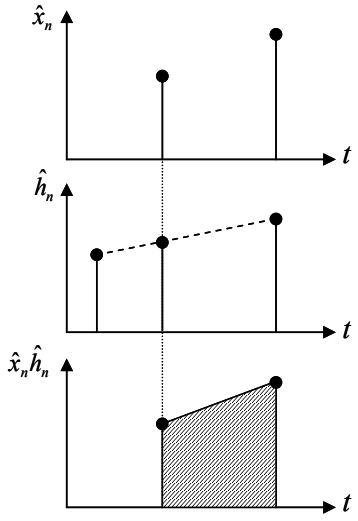


Figure 68 : Produit de convolution asynchrone d'ordre 1s

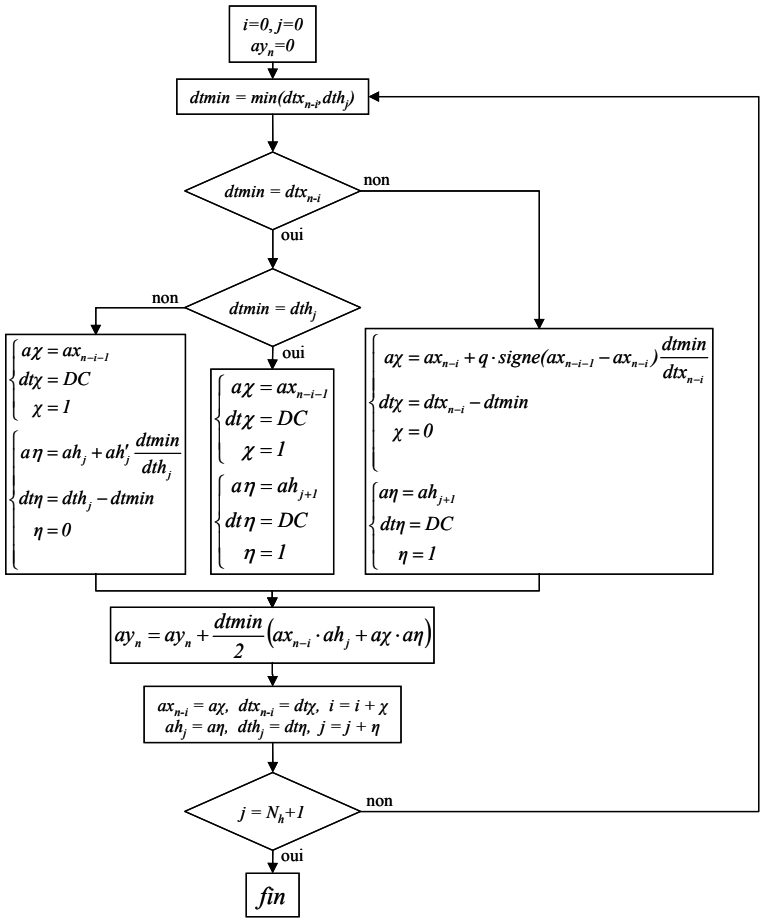


Figure 69 : Organigramme du produit de convolution asynchrone d'ordre 1s

V.2.2.12.2 Algorithme complet

Avec le calcul complet de l'aire, l'amplitude du signal de sortie correspond à la somme des surfaces du produit des fonctions interpolées. Ces deux fonctions étant linéaires, leur produit est une fonction du second ordre. Le calcul de l'intégrale complète correspond donc l'intégration d'un polynôme d'ordre 3. Le principe de cette technique est présenté sur la Figure 70.

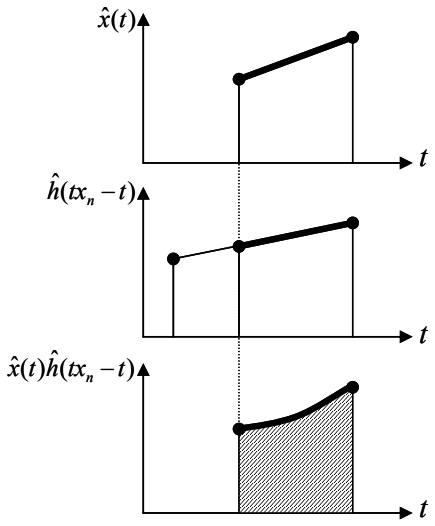


Figure 70 : Produit de convolution asynchrone d'ordre 1c

Comme il n'y a encore qu'une seule configuration, l'algorithme reste relativement simple compte tenu de la complexité amenée par le calcul complet de l'intégrale. La structure de base est donc la même qu'avec la méthode simplifiée. La différence vient du fait que le calcul de l'aire ne dépend que plus explicitement des points suivants mais uniquement des amplitudes courantes et des pentes (appelées respectivement *px* et *ph* pour les signaux *x* et *h*) :

$$px = \frac{ax_{n-i} - ax_{n-i-1}}{dtx_{n-i}} = \pm\frac{q}{dtx_{n-i}} \text{ et } ph = \frac{ah_j - ah_{j+1}}{dth_j} = \frac{ah'_j}{dth_j} \qquad \text{Eq. (135)}$$

$$A = \frac{dtmin^3}{3} px \cdot ph + \frac{dtmin^2}{2}\left(ax_{n-i} \cdot ph + ah_j \cdot px\right) + dtmin \cdot ax_{n-i} \cdot ah_j \qquad \text{Eq. (136)}$$

Nous remarquons qu'en normalisant les pentes par *dtmin* nous simplifions le calcul de l'intégrale :

$$A = dtmin\left(\frac{px \cdot ph}{3} + \frac{1}{2}\left(ax_{n-i} \cdot ph + ah_j \cdot px\right) + ax_{n-i} \cdot ah_j\right) \qquad \text{Eq. (137)}$$

Finalement le produit de convolution asynchrone d'ordre 1c est réalisé en suivant l'organigramme présenté Figure 71

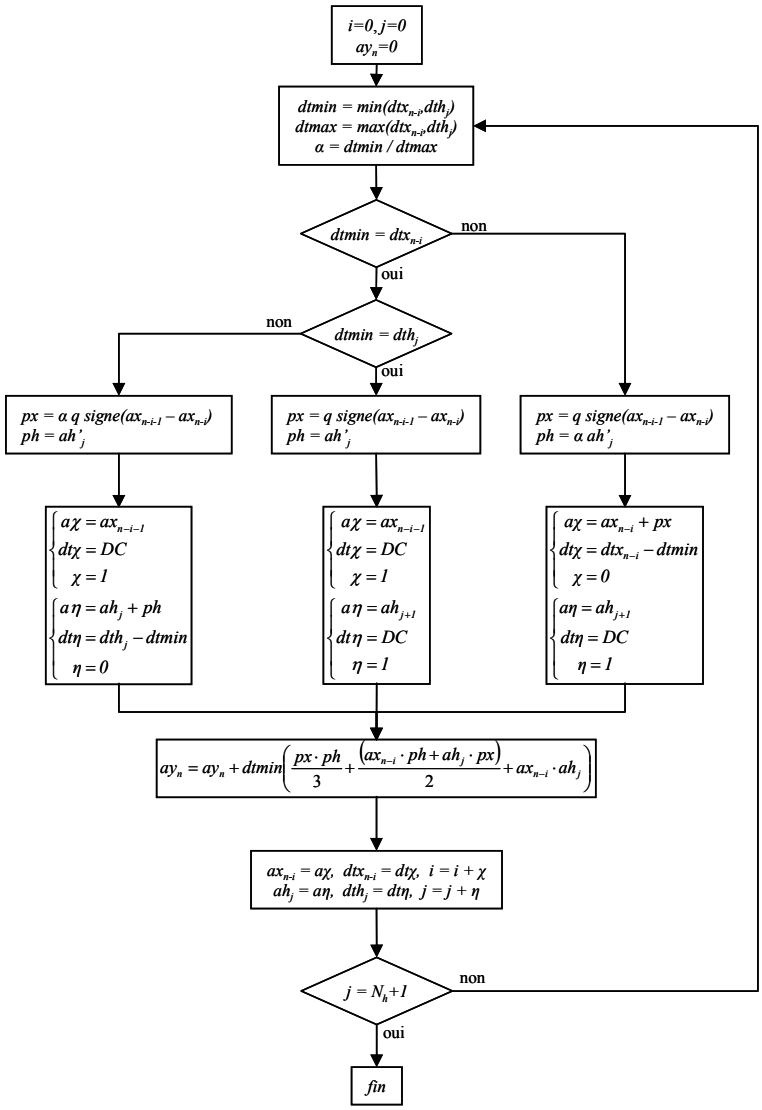


Figure 71 : Organigramme du produit de convolution asynchrone d'ordre 1c

V.2.2.12.3 Complexités combinatoires

La complexité combinatoire dépend de la technique utilisée pour calculer l'aire.

Avec la méthode simplifiée, il faut trois multiplications et deux additions pour calculer la surface puis deux multiplications et deux additions pour mettre à jour les amplitudes et les intervalles de temps. Compte tenu des B_n itérations nécessaires au calcul complet de la sortie, nous pouvons en déduire que la complexité combinatoire du produit de convolution asynchrone d'ordre 1s est :

$$O_{1s,n} = 9B_n \qquad \text{Eq. (138)}$$

Avec la méthode complète, le nombre d'opérations nécessaires au calcul de la surface est augmenté : il faut en effet cinq multiplications et quatre additions pour calculer l'aire d'une plage. Cependant, il faut toujours deux multiplications et deux additions pour mettre à jour les amplitudes et les intervalles de temps. Au total, la complexité combinatoire du produit de convolution asynchrone d'ordre 1c vaut :

$$O_{1c,n} = 13B_n \qquad \text{Eq. (139)}$$

V.2.2.12.4 Erreurs de calcul

Nous représentons sur la Figure 72, l'étude du pire cas pour le produit de convolution asynchrone d'ordre 1 utilisant l'algorithme simplifié et complet. Avec les deux méthodes, nous pouvons remarquer que le pire cas est encore défini par l'aire du rectangle inférieur (équation (119)).

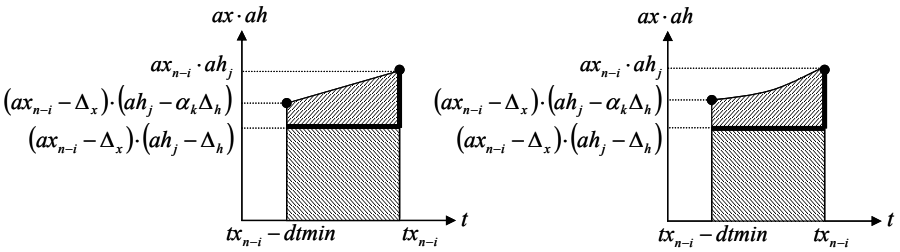


Figure 72 : Etude du pire cas de l'ordre 1

Avec la méthode simplifiée, l'aire calculée en fonction de α_k vaut :

$$A_{1s} = \frac{dtmin}{2} \cdot \left(ax_{n-i} \cdot ah_j + \left(ax_{n-i} - \Delta_x\right) \cdot \left(ah_j - \alpha_k \Delta_h\right)\right) \qquad \text{Eq. (140)}$$

Nous pouvons ensuite déterminer l'erreur relative de l'algorithme simplifié :

$$\varepsilon_{M1s} = \frac{1}{2} \cdot \left(\frac{ax_{n-i}}{ax_{n-i} - \Delta_x} \cdot \frac{ah_j}{ah_j - \Delta_h} - 1 + \frac{(1-\alpha_k)\Delta_h}{ah_j - \Delta_h} \right) \qquad \text{Eq. (141)}$$

Finalement, compte tenu de sa valeur, ε_{M1s} est borné :

$$\frac{\varepsilon_{M0}}{2} \leq \varepsilon_{M1s} < \varepsilon_{M0} \qquad \text{Eq. (142)}$$

Avec la méthode complète, nous proposons de déterminer l'aire calculée en partant de l'équation (137). Pour respecter les hypothèses utilisées pour l'étude du pire cas, nous remplaçons les pentes normalisées *px* et *ph* par $-\Delta_x$ et $-\alpha_k\Delta_h$. Nous obtenons ainsi l'aire suivante :

$$A_{1c} = dtmin\left(ax_{n-i} \cdot ah_j - \frac{1}{2}\left(ax_{n-i} \cdot \Delta_h + \alpha_k \cdot ah_j \cdot \Delta_x\right) - \frac{\alpha_k \Delta_x \Delta_h}{3} \right) \qquad \text{Eq. (143)}$$

L'erreur relative de l'algorithme complet est alors défini par :

$$\varepsilon_{M1s} = \frac{1}{2}\left(\frac{ax_{n-i}\left(ah_j - \alpha_k \Delta_h\right)}{ax_{n-i} - \Delta_x} + \frac{ah_j}{ah_j - \Delta_h} \right) + \frac{\alpha_k}{3} \frac{\Delta_x}{ax_{n-i} - \Delta_x} \cdot \frac{\Delta_h}{ah_j - \Delta_h} \qquad \text{Eq. (144)}$$

Pour simplifier son interprétation, nous bornons ε_{M1s} par :

$$\frac{\varepsilon_{M0}}{2} < \varepsilon_{M1c} < \varepsilon_{M0} \qquad \text{Eq. (145)}$$

V.2.2.13 Conclusion : comparaison des méthodes

Nous venons d'étudier un nouveau type de produit de convolution de signaux échantillonnés. Il est fondé sur le principe d'un produit de convolution analogique des signaux interpolés. Pour effectuer l'interpolation et le calcul de l'aire du produit des fonctions, nous avons proposé trois interpolations (ordre 0, ordre 0.5 *au point le plus proche* et ordre 1) et deux méthodes de calcul de l'aire. La première, appelée *simplifiée*, utilise une approximation de l'intégrale (rectangle pour l'ordre 0, trapèze pour les autres) tandis que la seconde appelée *complète* calcule l'intégrale exacte du produit des fonctions. Au final, nous avons décrit cinq produits de convolution différents, puis étudié leurs complexités combinatoires et enfin exprimé les erreurs relatives maximales qu'ils pouvaient commettre.

Nous proposons maintenant de déterminer les interpolations optimales pour lesquelles le compromis entre la complexité combinatoire et les erreurs commises est le plus *acceptable*. Dans la mesure où l'erreur dépend des signaux (deux paramètres pour chaque signal), elle ne peut pas

être quantifiée; nous proposons donc de les normaliser par rapport à celle de l'ordre 0. Le Tableau 4 récapitule en fonction des méthodes l'ensemble des études menées dans les sections précédentes en terme de complexité et d'erreurs. Il rappelle également le nombre de tests effectués et le nombre de variables locales utilisées :

	Ordre 0	**Ordre 0.5 simplifié**	**Ordre 0.5 complet**	**Ordre 1 simplifié**	**Ordre 1 complet**
Complexité	4	6	13	9	13
Erreur relative maximale	1	0.5 ou 0.75	[0.5 ; 0.75]	[0.5 ; 1[	]0.5 ; 1[
Nbe de tests	3	5	6	3	3
Nbe de variables locales	0	6	9	6	6

Tableau 4 : Comparaison des caractéristiques des différents produits de convolution asynchrone

La Figure 73 représente l'erreur relative maximale normalisée en fonction de la complexité combinatoire. Nous pouvons remarquer que les méthodes d'ordre 0 et d'ordre 0.5s offre les meilleurs compromis entre les deux caractéristiques.

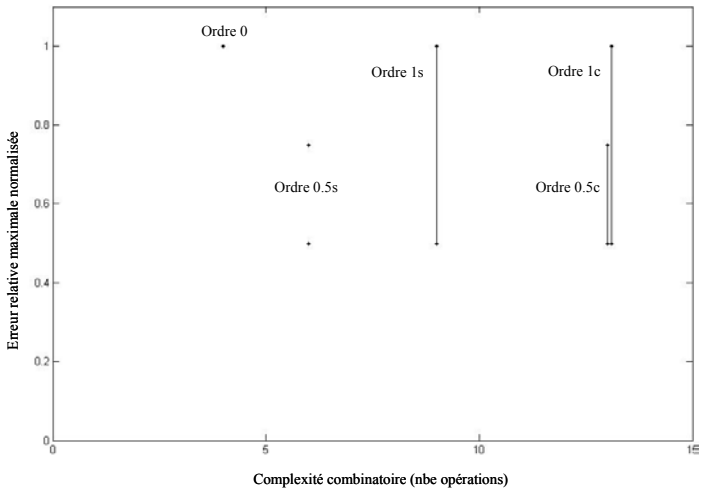


Figure 73 : Représentation de l'erreur relative maximale normalisée en fonction de la complexité combinatoire pour chaque méthode

V.3 Filtre numérique RIF asynchrone

V.3.1 Réponse impulsionnelle

La sortie d'un filtre numérique à réponse impulsionnelle finie est calculée à partir du produit de convolution entre le signal d'entrée échantillonné et la réponse impulsionnelle finie. Nous avons vu précédemment qu'il était possible de faire un produit de convolution entre deux signaux échantillonnés non uniformément. Comme il est évident qu'un signal échantillonné uniformément est un cas particulier d'un échantillonnage non uniforme où tous les intervalles de temps sont constants, il est donc possible de calculer le produit de convolution entre le signal d'entrée échantillonné non uniformément et la réponse impulsionnelle d'un filtre quelconque. Ceci présente l'énorme avantage de ne pas avoir à inventer des techniques de conception de filtres à réponses impulsionnelles échantillonnées non uniformément. Il suffit simplement d'utiliser les techniques habituelles de conception comme la méthode des fenêtres, la méthode de l'échantillonnage en fréquence ou des méthodes optimales par exemple.

A partir de maintenant nous allons considérer la fonction h comme un signal échantillonné régulièrement à la fréquence $F_e = 1/T_e$.

V.3.2 Complexité combinatoire

La complexité combinatoire dépendant d'une part du choix des interpolations utilisées dans l'algorithme et d'autre part du nombre de boucles effectuées B_n. Sachant que la réponse impulsionnelle h est échantillonnée régulièrement l'estimation du nombre de boucles peut se simplifier :

$$B_n \approx N_h \left(1 + \frac{T_e}{moy(dtx|_{[tx_n - T_h ; tx_n]})} \right) \qquad \text{Eq. (146)}$$

Le nombre de boucle B_n pour calculer la nième sortie dépend des intervalles de temps du signal d'entrée sur la plage $[tx_n\text{-}T_h;\ tx_n]$ et de la valeur de la période d'échantillonnage de la réponse impulsionnelle. En fonction de leur rapport, B_n peut prendre plusieurs ordres de grandeur.

Si les intervalles de temps sont très grands devant T_e alors le rapport tend vers 0; donc B_n tend vers N_h. Cela veut dire qu'en réalité le nombre d'échantillons utilisés du signal d'entrée est très faible et donc que la valeur de sortie est peu précise.

Si les intervalles de temps sont de l'ordre de T_e alors le rapport tend vers 1; B_n tend donc vers $2N_h$. Le nombre d'échantillons d'entrée est a peu près égal au nombre de coefficients de la réponse impulsionnelle.

Si les intervalles de temps sont très petits devant T_e alors B_n sera beaucoup plus grand que $2N_h$. Le nombre d'échantillons d'entrée est alors fortement supérieur au nombre de coefficients mais la valeur de la sortie est plus précise.

V.3.3 Réponse en fréquence

L'avantage du produit de convolution asynchrone est que le filtrage d'un signal échantillonné non uniformément est linéaire malgré l'approximation nécessaire lors du calcul puisqu'il est défini à temps continu. Pour une méthode donnée, nous pouvons donc étudier le filtrage dans le plan fréquentiel.

Dans le cas idéal, le filtrage s'écrit :

$$Y(f) = H(f)X(f) \qquad \text{Eq. (147)}$$

Or, d'un côté, le signal d'entrée est reconstruit selon une des méthodes présentées dans le chapitre précédent introduisant une distorsion que l'on notera *D(f)*. Et de l'autre, la réponse impulsionnelle est également reconstruite à partir d'une fonction unique *W(f)*. Le résultat du produit de convolution s'écrit donc :

$$\hat{Y}(f) = \hat{H}(f)\hat{X}(f) \text{ avec } \hat{H}(f) = W(f)H(f) \text{ et } \hat{X}(f) = X(f) + D(f) \qquad \text{Eq. (148)}$$

L'erreur relative de filtrage peut alors être exprimée en fonction des paramètres d'entrée :

$$\varepsilon(f) = \frac{\hat{Y}(f) - Y(f)}{Y(f)} = \left(1 - W(f)\right) + W(f)\frac{D(f)}{X(f)} \qquad \text{Eq. (149)}$$

Comme nous l'avons montré précédemment l'erreur de filtrage ne dépend pas du filtre mais des techniques d'interpolation. Le terme de gauche *(1-W(f))* représente en module l'atténuation du

filtre dans la bande de base apportée par la reconstruction de la réponse impulsionnelle. Il tend vers 0 quand *W(f)* tend vers un filtre de reconstruction passe bas idéal. Le terme de droite correspond aux composantes haute fréquence générées par la distorsion qui ne sont pas filtrées du fait du caractère non idéal du filtre passe bas de reconstruction de la réponse impulsionnelle. Pour diminuer l'erreur dans la bande passante, une solution serait de prédéformer le gabarit la réponse en fréquence du filtre afin d'annuler les effets de la reconstruction. Par ailleurs, comme dans tous les cas, la fenêtre *W(f)* est un sinus cardinal (ou un sinus cardinal au carré pour l'interpolation d'ordre 1) d'amplitude maximale T_e. L'utilisation du produit de convolution basé sur une interpolation introduit une atténuation qu'il faut compenser. Ainsi la réponse en fréquence initiale du filtre doit être amplifié d'un facteur F_e.

En revanche, la phase de la réponse en fréquence est irrémédiablement modifiée par la reconstruction. En effet, l'interpolation de deux signaux introduit un déphasage dû au décalage de la fonction de reconstruction pour rendre le système causal. Pour la réponse impulsionnelle, le décalage étant constant (une demi période d'échantillonnage pour l'ordre 0 et une période pour l'ordre 0,5 et 1), la phase apportée par la reconstruction est linéaire. Le temps de propagation de groupe est alors constant et correspond simplement au retard du signal reconstruit sur le signal original. En revanche pour le signal échantillonné non uniformément, le décalage n'est plus constant puisque égal à chaque intervalle de temps. La phase du signal reconstruit n'est alors plus linéaire. Par conséquent si la réponse en fréquence du filtre est linéaire, le déphasage entre le signal d'entrée et de sortie ne sera plus linéaire. Cependant en observant un signal reconstruit, nous pouvons facilement constater que le temps de propagation de groupe, à défaut d'être constant, peut être approché par la moyenne des intervalles de temps linéarisant la phase à une erreur $\varphi_D(f)$ près, liée à la distorsion introduite par la reconstruction non uniforme :

$$\arg\left(\frac{\hat{Y}(f)}{X(f)}\right) = -(N_h - 1)\pi f T_e - (1+\sigma)\pi f T_e - moy(dtx)\pi f + \varphi_D(f) \qquad \text{Eq. (150)}$$

où N_h-*1* est l'ordre du filtre et σ un coefficient différentiant les interpolations au voisin le plus proche de et d'ordre 1 ($\sigma = 1$) du bloqueur d'ordre 0 ($\sigma = 0$); en effet, les interpolations n'induisent pas le même retard, donc la même phase lors de la reconstruction. L'erreur de phase relative vaut alors :

$$\varepsilon_{\varphi}(f) = \frac{\arg\left(\frac{\hat{Y}(f)}{X(f)}\right) - \arg\left(\frac{Y(f)}{X(f)}\right)}{\arg\left(\frac{Y(f)}{X(f)}\right)} = \frac{1+\sigma}{N_h - 1} + \frac{moy(dtx)}{T_e}\frac{1}{N_h - 1} + \frac{\varphi_D(f)}{(N_h - 1)\pi f T_e} \quad \text{Eq. (151)}$$

L'équation (150) montre que si l'ordre du filtre est réduit de $1 + \sigma$ par rapport à celui donnant la phase souhaitée alors l'effet de la reconstruction de la réponse impulsionnelle est annulé. En revanche, en ce qui concerne l'effet de la reconstruction du signal d'entrée, il y a deux options. Si la moyenne des intervalles est de l'ordre de grandeur de la période d'échantillonnage : dans ce cas, nous savons que la fréquence d'échantillonnage est grande devant la bande utile du signal d'entrée car l'échantillonnage par traversée de niveaux suréchantillonne le signal d'entrée quand il varie. Donc pour assurer une bonne sélectivité du filtre, l'ordre du filtre doit être augmenté ce qui réduit le terme central de l'équation (151). Si en moyenne, les intervalles de temps sont beaucoup plus petits que T_e et le terme central est directement négligeable. Par ailleurs, dans tous les cas, plus l'ordre du filtre est grand, plus le terme de droite est diminué et plus la linéarité est respectée. La phase n'est donc plus linéaire mais *quasiment* linéaire.

V.3.4 Exemples de filtrage

Nous proposons dans cette section d'illustrer le filtrage RIF de signaux échantillonnés non uniformément selon les méthodes présentées précédemment. Afin d'être synthétique, nous nous limiterons à ne présenter que les filtrages passe-bas et passe-haut (les filtrages passe-bande et coupe bande étant des combinaisons des autres) d'un signal périodique à deux composantes : une fréquence fondamentale de 1Hz et son harmonique 5 à 5Hz. Le signal est échantillonné sur 15 niveaux répartis régulièrement sur sa dynamique. Les filtres ont une réponse impulsionnelle à 33 coefficients (ordre 32) échantillonnée uniformément à la fréquence $F_e = 20Hz$. La fréquence de coupure est fixée à $f_c = 3Hz$ et les coefficients sont obtenus par la méthode des fenêtres [Prado 2000] (fenêtre de Hamming). Les résultats sont représentés sur les Figure 74 et Figure 75.

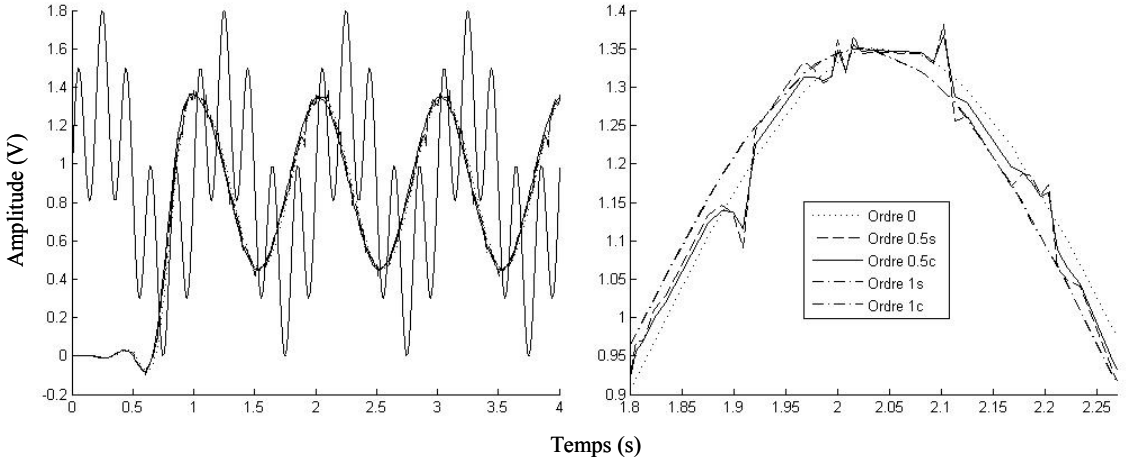


Figure 74 : Filtrage RIF passe-bas à 33 coefficients de fréquence de coupure $f_c = 3Hz$ et échantillonnés à $F_e = 20Hz$ selon les cinq interpolations (plan large et zoom)

A première vue, nous pouvons constater que le signal d'entrée est correctement filtré puisque, dans un cas, l'harmonique 5 est éliminée (Figure 74) tandis que dans l'autre, la composante continue et la fréquence fondamentale sont filtrées (Figure 75). En revanche, en zoomant pour observer les différentes méthodes, nous remarquons que l'utilisation d'une interpolation au voisin le plus proche entraîne des oscillations, dans le calcul d'une sortie à l'autre, qui sont amplifiées lors du filtrage passe-haut – oscillations qui n'apparaissent pas avec les interpolations d'ordre 0 et d'ordre 1. La cause est intrinsèque à l'interpolation au voisin le plus proche. En effet, quelle que soit l'interpolation, le filtrage repose sur le produit de convolution asynchrone, c'est-à-dire une décomposition en aires élémentaires. Chaque nouvel échantillon d'entrée implique un nouvel échantillon en sortie, donc une nouvelle décomposition. Le bloqueur d'ordre 0 utilise toujours un seul échantillon de chaque signal pour le calcul d'une aire élémentaire et le bloqueur d'ordre 1 en utilise toujours deux (un point fixe et un point interpolé) alors que l'interpolation au voisin le plus proche en utilisent parfois un ou parfois deux, en fonction du rapport entre les intervalles de temps des deux signaux. Ainsi, d'une sortie à l'autre, le nombre total d'échantillons utilisés, pour la méthode simplifiée 0.5s illustrée Figure 62, ou le nombre d'aires totale utilisées, pour la méthode complète 0.5c illustrée Figure 64, varie. Si la différence d'amplitude entre deux points successifs est grande, le fait de choisir, ou pas, le point suivant à une grande influence sur la valeur de sortie. En revanche, plus la fréquence d'échantillonnage est grande, plus l'influence est petite car d'une part, la réponse impulsionnelle a systématiquement l'intervalle de temps le plus petit et d'autre part, le nombre de points utilisés ou le nombre d'aire totale devient constant d'une sortie à l'autre. Ainsi, nous avons représenté sur la Figure 76, le nombre total d'aires utilisées au cours du temps par la méthode 0.5c. Nous avons normalisé les

résultats par le nombre d'aires élémentaires dans chaque cas, pour remettre à l'échelle, sachant qu'il peut y avoir deux ou trois surfaces dans une aire élémentaire.

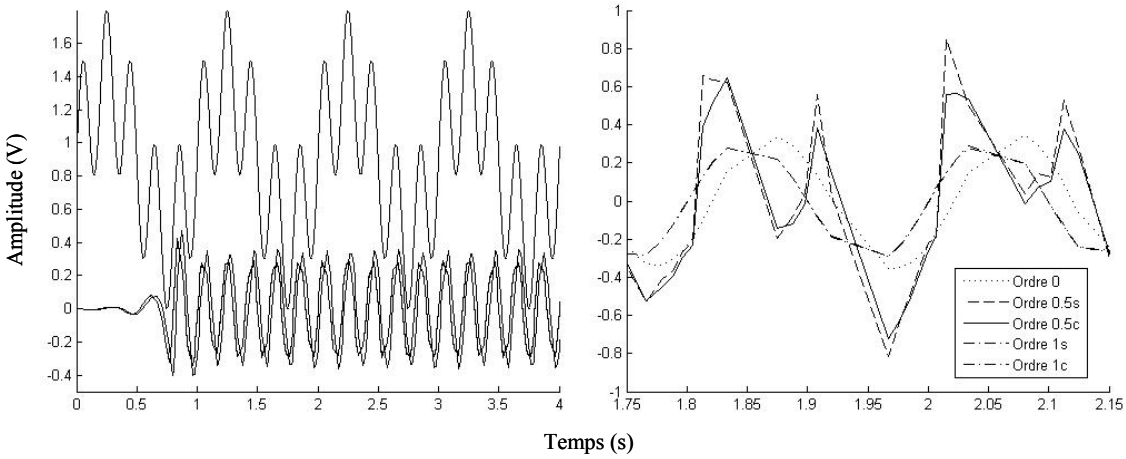


Figure 75 : Filtrage RIF passe-haut à 33 coefficients de fréquence de coupure $f_c = 3Hz$ et échantillonnés à $F_e = 20Hz$ selon les cinq interpolations (plan large et zoom)

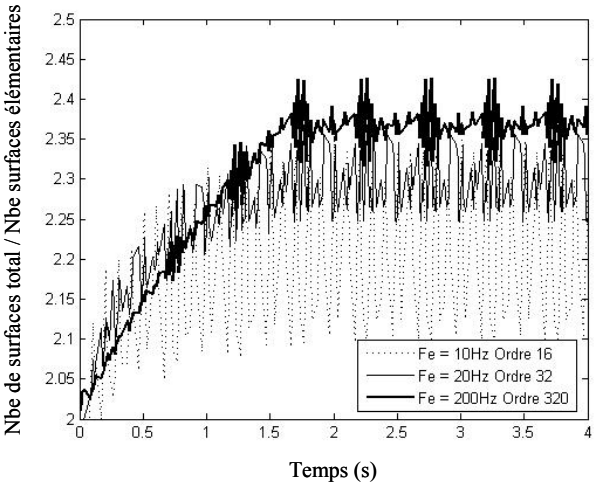


Figure 76 : Evolution du nombre de surfaces totales dans le produit de convolution asynchrone d'ordre 0.5c en fonction de la fréquence d'échantillonnage de la réponse impulsionnelle

Par ailleurs si les oscillations lors du filtrage passe-haut sont plus importantes que lors du filtrage passe-bas, cela est dû au fait que la réponse impulsionnelle du filtre passe haut à un saut d'amplitude important au niveau de l'échantillon central comme montre la figure suivante :

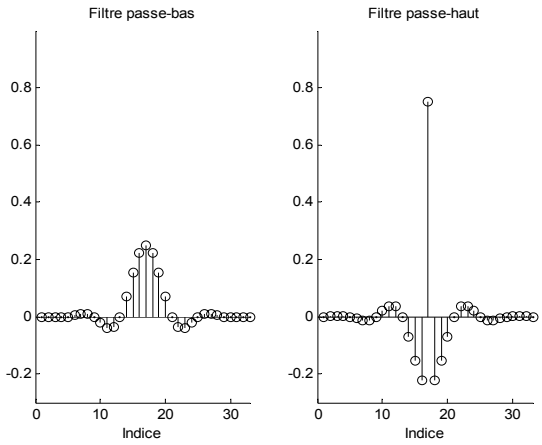


Figure 77 : Réponses impulsionnelles des filtres passe-bas et passe-haut

Nous proposons un second exemple pour illustrer les modifications apportées par l'interpolation sur un filtre à phase linéaire. Pour cela, nous simulons le filtrage d'un signal sinusoïdal pur par un filtre dont la fréquence de coupure est supérieure à la fréquence du signal afin qu'il puisse simplement être retardé. Nous créons deux filtres ayant des caractéristiques différentes mais un temps propagation de groupe identique : un filtre d'ordre 16 et de fréquence d'échantillonnage $F_e = 10Hz$ et un filtre d'ordre 32 et de fréquence d'échantillonnage $F_e = 20Hz$. Le temps de groupe généré par la phase linéaire est égal à 1,6s dans les deux cas. Compte tenu des remarques précédentes sur le produit de convolution d'ordre 0.5, nous ne simulons que les méthodes d'ordre 0 et 1. Les résultats sont présentés Figure 78.

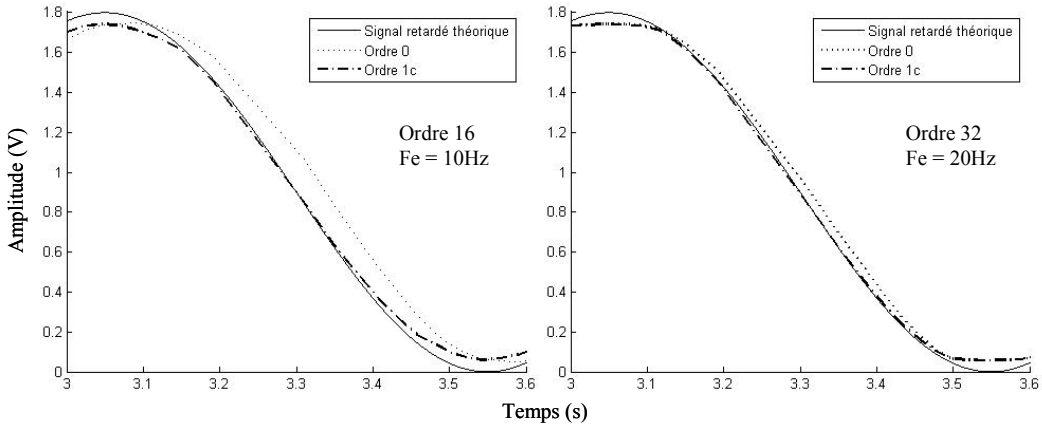


Figure 78 : Filtrages à phase linéaire d'un signal sinusoïdal

On peut noter qu'à l'ordre 1, le temps de propagation du filtrage est approximativement égal au temps de groupe théorique. En revanche, celui apporté à l'ordre 0 est beaucoup plus grand. Sur la partie droite de la figure, nous pouvons observer qu'en augmentant l'ordre et la fréquence

d'échantillonnage, il tend vers le temps théorique conformément aux remarques effectués dans la section précédente.

En conclusion, l'étude théorique nous permet de montrer que seul un filtrage RIF utilisant un produit de convolution d'ordre 0 ou 0.5 ne peut être raisonnablement implémenté compte tenu notamment de la complexité combinatoire; les simulations pratiques du filtrage nous montrent que le produit d'ordre 0.5 n'est pas assez robuste pour supporter les écarts de calculs d'une sortie à l'autre. Nous proposons donc d'étudier l'implémentation matérielle d'un filtre RIF à partir d'un produit de convolution d'ordre 0.

V.4 Architecture matérielle

V.4.1 Choix de la structure

La structure implémentant l'algorithme de convolution asynchrone doit être suffisamment versatile pour supporter les variations du nombre d'opérations dans le calcul d'une sortie à l'autre. Une structure directe par exemple ne respecte pas ce critère car le nombre de retard, de multiplieurs serait trop compliqué à commander en fonction des intervalles de temps de l'entrée. La solution que nous proposons est l'utilisation d'une structure itérative représentant l'image d'une boucle de l'algorithme. L'architecture proposée est décrite au niveau conception.

V.4.2 Structure itérative dans le cas uniforme

Dans le cas standard, la structure itérative est utilisée pour minimiser le nombre de multiplieurs dans le circuit et donc sa surface. Ce choix est généralement fait au détriment de la vitesse car il faut alors N_h cycles pour calculer un seul échantillon de sortie. La structure possède un multiplieur (*MULT*), un accumulateur (*ACC*), un registre à décalage (*RD*) pour stocker les échantillons d'entrée, une ROM contenant en mémoire les coefficients de la réponse impulsionnelle, et un buffer (*BUFF*) pour valider la sortie après les N_h cycles. L'ensemble du système est contrôlé par une machine à état (*FSM*) pilotant les différents blocs en synchronisme avec un signal d'horloge.

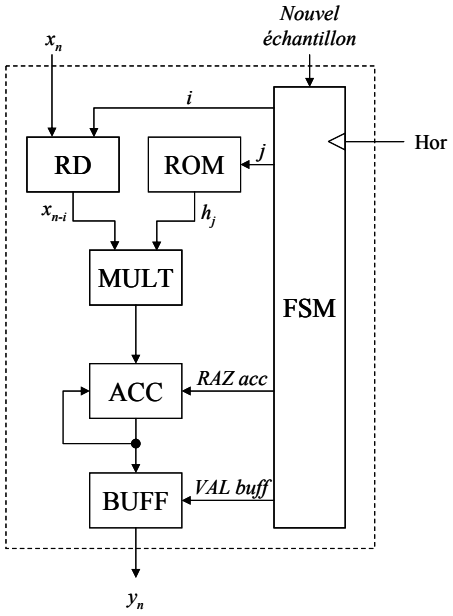


Figure 79 : Structure itérative d'un filtre RIF dans le cas uniforme

V.4.3 Structure itérative dans le cas non uniforme

Dans le cas non uniforme, la structure itérative, décrite précédemment, peut être modifiée pour implémenter le produit de convolution asynchrone d'ordre 0. L'architecture générale est présentée Figure 80 [Aeschlimann *et al.* 2004]. On peut remarquer qu'il y a quatre registres à décalage au lieu d'un seul. En effet, deux sont destinés à stocker les amplitudes et les intervalles de temps des échantillons d'entrée (*RD ax* et *RD dtx*) tandis que les deux autres remplacent la ROM pour mémoriser les coefficients de la réponse impulsionnelle et ses intervalles de temps (*RD ah* et *RD dth*). Bien qu'a priori, il n'y ait qu'un seul intervalle de temps pour la réponse impulsionnelle, nous avons choisi de présenter une architecture la plus générale possible pouvant utiliser des réponses impulsionnelles échantillonnées non uniformément.

Par ailleurs, la structure comporte deux multiplieurs (soit un de plus que dans le cas uniforme), un accumulateur (*ACC*) et un buffer (*BUFF*); le flot de calcul étant le même que dans la structure classique. Enfin l'architecture du filtre non uniforme introduit trois nouveaux blocs : la mise à jour des intervalles de temps (*MAJ*) et des adresses des registres (*ADD*), et le calcul de l'intervalle de temps minimum (*MIN*). Il n'y a plus de machine à état pour piloter les blocs mais un

contrôle local asynchrone (qui ne figure pas sur la figure pour des raisons de clarté – chaque bloc ayant son propre contrôleur connecté aux autres blocs avec lesquels il communique).

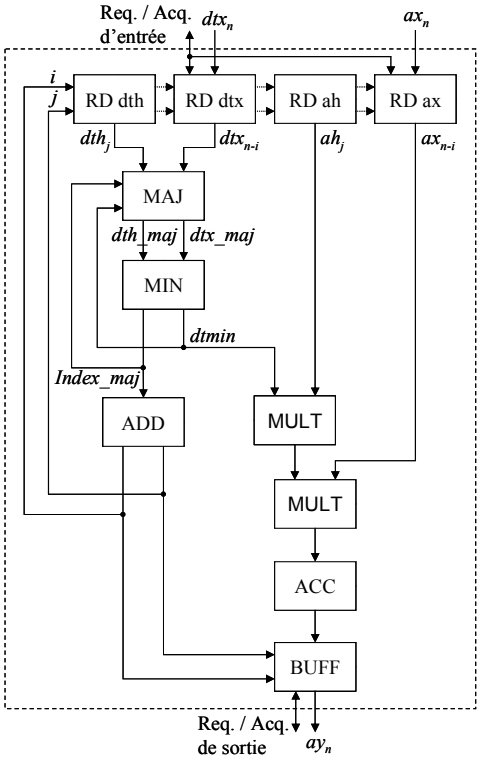


Figure 80 : Structure itérative d'un filtre RIF dans le cas non uniforme

V.4.3.1 Registre à décalage

Le registre à décalage est une structure linéaire d'élément mémorisant stockant les données en entrées (Figure 81). Lorsqu'une requête arrive en entrée (*Req. entrée*), les données précédentes se déplacent d'un cran tandis que la nouvelle donnée est mémorisée dans le premier élément. Lorsque toutes les données sont figées un acquittement est renvoyé (*Acq. entrée*). Une nouvelle valeur de sortie peut alors être calculée. Le registre reçoit alors une requête de l'étage *ADD* (*Req. ADD*) pour lui indiquer qu'une nouvelle adresse est disponible. Le registre peut alors récupérer l'adresse, acquitter le bloc *ADD* (*acq. ADD*) puis placer en sortie la donnée correspondante grâce à un multiplexeur et envoyer une requête à l'étage suivant (*Req. MULT* pour *RD ax* et *Req. MAJ* pour *RD dtx*).

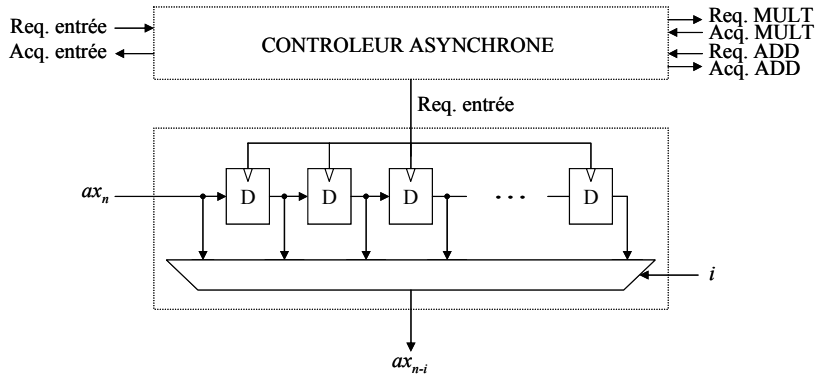


Figure 81 : Architecture du bloc *RD ax* : registre à décalage des amplitudes du signal d'entrée

Pour les registres de la réponse impulsionnelle, le fonctionnement est le même sauf qu'il n'y a ni signaux de requête et d'acquittement, ni nouvelle valeur en entrée. Ainsi, les données du registre ne sont jamais décalées, il se comporte simplement comme une mémoire.

Lors de la conception du filtre, il faut spécifier la profondeur du registre, c'est-à-dire le nombre maximal de données qu'il peut stocker. Pour les registres de la réponse impulsionnelle, leur taille correspond directement au nombre de coefficients. En revanche, pour le signal d'entrée, étant donné que le nombre d'échantillons utilisés varie pour le calcul d'une sortie, la taille du registre doit prendre en compte cette fluctuation. A partir des caractéristiques du filtre, il est possible de déterminer une borne supérieure : en effet, pour un filtre à N coefficients, échantillonné uniformément à la fréquence $F_e = 1/T_e$, le nombre maximal d'échantillons d'entrée est obtenu lorsque tous les intervalles de temps sont minima (donc égaux au temps de boucle du convertisseur δ) pendant la durée du filtrage i.e. NT_e :

$$M_{max} = N \frac{T_e}{\min(dtx_{n-i})} = N \frac{T_e}{\delta} \qquad \text{Eq. (152)}$$

Or pour un signal analogique $x(t)$ à bande limitée de fréquence maximale F_{max} et de dynamique V_{alim}, nous savons que le temps de boucle du convertisseur peut s'exprimer en fonction de x d'après l'inégalité de Bernstein [Delmas 1991] :

$$\left| \frac{dx(t)}{dt} \right| \leq \frac{q}{\delta} = \pi V_{alim} F_{max} \qquad \text{Eq. (153)}$$

Comme la fréquence maximale du signal d'entrée est au moins deux fois plus petite que la fréquence d'échantillonnage :

$$F_{max} = \frac{F_e}{k} \leq \frac{F_e}{2} \text{ pour tout } k \geq 2 \qquad \text{Eq. (154)}$$

Le nombre maximal d'échantillons d'entrée devient :

$$M_{max} = N \frac{\pi}{k} \frac{V_{alim}}{q} \leq \frac{\pi}{2} NL \qquad \text{Eq. (155)}$$

Ainsi pour un signal échantillonné sur L niveaux, il faut utiliser des registres à décalage dont la taille est environ L fois celle requise pour un filtre à N coefficients. En pratique ce cas n'est jamais atteint compte tenu de sa faible probabilité d'apparition, un registre 3 à 4 fois plus grand que le nombre de coefficient est suffisant. De plus, ce problème peut être simplifié par une implémentation logicielle à base de microcontrôleurs [Abrial *et al.* 2001] ou de microprocesseurs [Renaudin *et al.* 1999].

V.4.3.2 Mise à jour des données

Le produit de convolution asynchrone nécessite de mettre à jour les amplitudes et les intervalles de temps du signal d'entrée et de la réponse impulsionnelle. Le choix d'une architecture basée sur le produit à l'ordre 0 permet de restreindre la mise à jour des intervalles de temps. Pour simplifier la gestion du flot de données entre les intervalles de temps stockés dans les registres et le calcul de l'intervalle de temps minimum, le bloc *MAJ* a été inséré entre les blocs *RD dtx*, *RD dth* et le bloc *MIN*.

L'architecture du bloc *MAJ* est présentée Figure 82. Ainsi, à chaque nouvelle itération, les registres envoient une requête (*Req. RD dtx* et *Req. RD dth*) pour transmettre leurs intervalles de temps respectifs (dtx_{n-i} et et dth_j). En fonction du signal *Index_maj*, indiquant qui était l'intervalle le plus petit au cours de l'itération précédente, le bloc *MAJ* choisit de conserver à l'aide de multiplexeurs soit le nouvel intervalle entrant soit l'intervalle mis à jour par la valeur *dtmin*, parallèlement pour le signal d'entrée et la réponse impulsionnelle. Les deux intervalles de temps maintenant mis à jour sont sauvegardés jusqu'à l'itération suivante et envoyés au bloc *MIN* après émission d'une requête (*Req. MIN*). Le bloc *MIN* peut alors acquitter (*Acq. MIN*) et calculer le

nouvel intervalle minimum, puis envoyer une requête (*Req. MIN*) afin de transmettre le nouveau *dtmin* et le signal *Index_maj*.

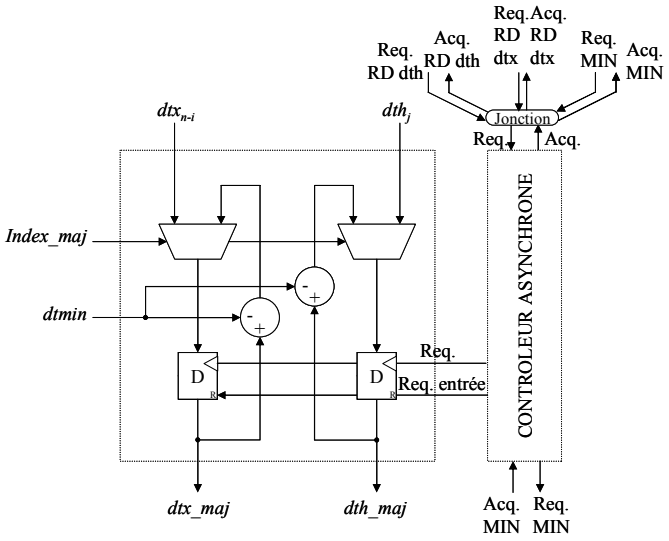


Figure 82 : Architecture du bloc *MAJ* : mise à jour des intervalles de temps

V.4.3.3 Adressage des registres

Parallèlement à la mise à jour des intervalles de temps, les adresses des registres doivent également être incrémentées à chaque itération. L'architecture du bloc *ADD* adressant les registres est décrite Figure 83. Lorsque l'intervalle de temps minimum a été calculé, le bloc *MIN* envoie une requête (*Req. MIN*). Le bloc *ADD* peut alors recevoir le signal *Index_maj* et acquitter (*Acq. MIN*). En fonction de la valeur de *Index_maj*, les valeurs des adresses *i* et *j* stockées depuis l'itération précédente sont mises à jour c'est-à-dire soit incrémentées de 1 soit de 0, valeurs choisies grâce aux multiplexeurs. Une requête est alors envoyée à tous les registres afin de transmettre la nouvelle adresse. Il est important de noter que même si la valeur d'une des deux adresses n'est pas modifiée, celle-ci doit quand même être envoyée afin que le registre concerné puisse acquitter et envoyer une requête aux blocs suivants.

Enfin, les adresses sont envoyées au bloc *BUFFER* pour valider la sortie (*Req. BUFFER*). Celui-ci détermine, à partir de la valeur des adresses, la condition de fin de boucle lorsque toutes les itérations ont été effectuées, c'est-à-dire quand tous les coefficients de la réponse impulsionnelle ont été utilisés : $j = N+1$.

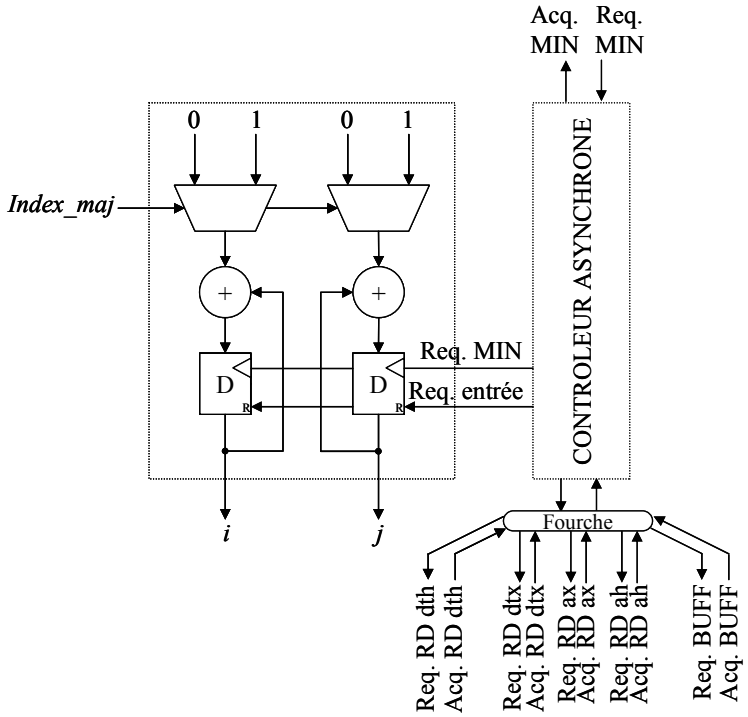


Figure 83 : Architecture du bloc *ADD* : mise à jour des adresses des registres

V.4.3.4 Latence de l'architecture

Le bon fonctionnement du circuit est assuré par le contrôle local asynchrone. Comme nous l'avons évoqué dans le chapitre I, un circuit asynchrone travaille en un temps minimum qui dépend du protocole de communication entre les contrôleurs et de l'implémentation matérielle retenue. Cependant, si le temps nécessaire pour une itération peut être considéré comme constant (bien que cela ne soit pas le cas en réalité), le temps total pour calculer une sortie dépend du nombre d'itérations effectuées. La latence de l'architecture n'est donc pas constante. Comme il n'y a aucune opération sur les intervalles de temps, un traitement en continu d'un signal échantillonné introduirait des décalages en sortie et donc une distorsion du signal reconstruit. A défaut d'être supprimé, cet effet peut être réduit si le concepteur rajoute lors de la synthèse une contrainte sur la latence en imposant qu'elle soit au moins égale à la résolution du timer dans le pire cas. Ainsi l'effet serait assimilable à l'erreur due à la quantification temporelle.

V.5 Conclusion

Nous venons de présenter une nouvelle méthode de filtrage de signaux échantillonnés non uniformément à l'aide de réponse impulsionnelle finie. L'équation aux différences entre les échantillons d'entrée et les coefficients, définie dans le cadre d'un échantillonnage supposé régulier, ne peut plus être utilisée – les retards n'étant plus constant; parallèlement, le traitement à temps continu en quantifiant le signal d'entrée engendre des réponses en fréquence périodisées et n'offre aucune architecture facilement implémentable.

Par conséquent, en revenant à la définition à temps continu, nous avons défini un nouveau produit de convolution à temps discret dit *asynchrone*. Il consiste à calculer à partir de signaux échantillonnés non uniformément, un produit de convolution analogique à un instant donné en interpolant les signaux à temps discret, en les multipliant puis en calculant leur intégrale. Afin de ne pas expliciter l'interpolation, nous avons introduit un algorithme itératif qui décompose le résultat en une somme d'aires élémentaires obtenues directement à partir des échantillons. Nous proposons ainsi cinq produits différents en fonction de l'interpolation : à l'ordre 0, selon le voisin le plus proche que nous appelons également 0.5 (calcul simplifié et complet de l'intégrale), et enfin à l'ordre 1 (calcul simplifié et complet de l'intégrale). Nous étudions pour l'algorithme, l'erreur dans le pire cas, ainsi que la complexité combinatoire nécessaire à chaque itération. Il apparaît clairement qu'une implémentation matérielle réaliste repose sur le choix d'une interpolation simple – la complexité combinatoire augmentant beaucoup plus fortement avec le degré de l'interpolation que ne décroît l'erreur maximale.

Dans la deuxième partie, nous avons réalisé un filtre à réponse impulsionnelle finie à partir de signaux échantillonnés non uniformément à partir du produit de convolution asynchrone d'ordre 0 dans lequel l'un des deux signaux est échantillonné régulièrement. Bien que le traitement de données espacées irrégulièrement empêche la conception de filtres à phase linéaire (caractéristique couramment recherchée pour créer un retard de groupe constant) – les rendant à phase approximativement linéaire, nous montrons qu'il est possible de filtrer facilement un signal échantillonné non uniformément. L'architecture matérielle qui en découle est d'ailleurs relativement similaire à la structure dans le cas uniforme – des modifications ayant dû être ajoutées afin de traiter à la fois les amplitudes et les intervalles de temps, et de prendre en compte la nature asynchrone du circuit.

Ainsi, nous avons montré, pour un traitement particulier, le filtrage RIF, que l'amplitude de sortie dépend à la fois des amplitudes et des intervalles de temps des signaux. Ainsi, conformément aux remarques que nous faisions à la fin du chapitre II, le filtrage basé sur le produit de convolution asynchrone est particulièrement bien adapté au traitement des signaux échantillonnés par traversée de niveaux puisque l'information portée par les intervalles de temps est bien prise en compte.

CHAPITRE VI

Filtrage numérique à réponse impulsionnelle infinie

Dans le chapitre précédent, nous avons montré, lors d'un filtrage RIF, qu'une relation entre des échantillons d'entrée et des échantillons de sortie pouvait être établie en définissant un produit de convolution entre le signal d'entrée et la réponse impulsionnelle. Cette méthode est dédiée à l'utilisation de réponses impulsionnelles finies qui limitent la durée du calcul. En revanche, pour les filtres à réponses impulsionnelles infinies, la technique présentée n'est plus applicable.

La solution que nous présentons dans ce chapitre est basée sur la représentation d'un filtre analogique dans l'espace d'état et sa discrétisation par un schéma numérique. Cette technique est rappelée un premier temps. Puis, de nouveaux schémas numériques sont introduits; leur stabilité est alors étudiée.

Dans un second temps, nous nous intéressons à l'implémentation matérielle d'un filtre numérique à réponse impulsionnelle infinie. Nous comparons les complexités combinatoires de chaque méthode afin d'éliminer les schémas nécessitant les coûts les plus importants. Nous en déduisons alors deux architectures asynchrones différentes pour deux schémas distincts : la méthode bilinéaire, inconditionnellement stable et la méthode Runge-Kutta d'ordre 4, conditionnellement stable.

VI.1 Introduction

VI.1.1 Filtrage RII de signaux échantillonnés régulièrement

Un filtre à réponse impulsionnelle infinie est toujours conçu selon le principe suivant [Prado 2000] : dans un premier temps, en fonction du cahier des charges, un gabarit du module ou de la phase, de la réponse en fréquence est spécifié. Puis dans un second temps, la fonction de transfert d'un filtre analogique respectant le gabarit de départ est calculée à partir d'une fonction d'approximation telle Butterworth ou Chebyshev. La fonction de transfert du filtre numérique peut alors en être déduite après remplacement de la variable *p*, définissant le plan complexe de Laplace, par la variable *z*, définissant le plan complexe *Z*. Une équation récursive reliant les échantillons de sortie et d'entrée est ainsi obtenue permettant l'implémentation du filtre. Cependant pour un même filtre analogique, plusieurs filtres numériques aux performances variables peuvent être conçus en fonction du choix du schéma numérique.

VI.1.2 Mise en équations

Quel que soit le type d'échantillonnage, le point de départ sera toujours la fonction de transfert du filtre analogique *H(p)* car c'est au moment de la discrétisation que l'échantillonnage rentre en ligne de compte. La fonction *H(p)*, décrite par l'équation (156) est une fonction rationnelle de degré *N*. Pour généraliser la notation, le degré du numérateur *M* sera supposé égal au degré du dénominateur bien qu'il puisse lui être inférieur. Ainsi un filtre quelconque aura *N + 1* coefficients au dénominateur et *N + 1* coefficients au numérateur dont *N – M* coefficients nuls.

$$H(p) = \frac{S(p)}{E(p)} = \frac{\sum_{i=0}^{N} a_i p^i}{\sum_{i=0}^{N} b_i p^i} \qquad \text{Eq. (156)}$$

VI.1.3 Limitation dans le cas non uniforme

A partir de la fonction de transfert analogique *H(p),* une fonction de transfert numérique *H(z)* est calculée dont l'équation aux différences permet de relier la $n^{ième}$ sortie, sous-entendu à l'instant $t_n = nT_e$, aux sorties et aux entrées précédentes ainsi qu'à l'entrée courante. Or la variable *z* de la fonction de transfert numérique représente un retard pur d'une période d'échantillonnage T_e,

ce qui implique que les instants précédents t_{n-1}, t_{n-2}, ..., t_{n-N} ne peuvent être quelconques : ils doivent valoir respectivement $(n-1)T_e$, $(n-2)T_e$,..., $(n-N)T_e$. La fonction de transfert ne peut donc être discrétisée que si l'intervalle de temps entre deux échantillons est constant. Ainsi lorsque l'échantillonnage du signal d'entrée est non uniforme, il n'est pas possible de définir un filtre à réponse impulsionnelle infinie à partir de sa fonction de transfert analogique. Pour déterminer une relation récurrente entre les signaux d'entrée et de sortie, il faut obligatoirement revenir en amont de la fonction de transfert en discrétisant directement son équation différentielle. La difficulté est alors de remplacer chaque dérivée (première, seconde jusqu'à celle d'ordre N) par une approximation particulière, ce qui complique considérablement la conception lorsque N devient grand.

VI.2 Représentation d'un filtre analogique dans l'espace d'état

Pour simplifier la discrétisation, l'équation différentielle doit être formulée dans l'espace d'état ne faisant ainsi intervenir qu'une seule dérivée première. L'équation (157) décrit la représentation d'état d'un système analogique où $x(t)$, $e(t)$ et $s(t)$ sont respectivement le vecteur d'état et les signaux d'entrée et de sortie.

$$\begin{cases} \dfrac{dx(t)}{dt} = Ax(t) + Be(t) \\ s(t) = Cx(t) + De(t) \end{cases} \qquad \text{Eq. (157)}$$

VI.2.1 Vecteur d'état

Considérons les transformées de Laplace des signaux d'entrée et de sortie du système notés $E(p)$ et $S(p) = H(p)E(p)$ et posons un vecteur d'état de taille N noté dans le domaine de Laplace $X(p) = (X_0(p),...,X_{N-1}(p))^T$ où chaque élément est défini par :

$$X_k(p) = \frac{p^k}{\sum_{n=0}^{N} b_n p^n} E(p) \qquad \text{Eq. (158)}$$

La transformée de Laplace de la sortie du système s'écrit :

$$S(p) = \sum_{n=0}^{N} a_n X_n(p) \qquad \text{Eq. (159)}$$

Pour les indices n allant de *1* à *N-1*, chaque élément du vecteur d'état $X_n(p)$ peut être calculé en fonction du précédent : $X_n(p) = pX_{n-1}(p)$. En revanche l'équation (159) nécessite un $(N+1)^{ième}$ élément $X_N(p)$ non défini dans l'équation (158). Or en partant de $X_0(p)$, l'équation (158) peut s'écrire :

$$\sum_{n=0}^{N} b_n p^n X_0(p) = E(p) \qquad \text{Eq. (160)}$$

En décomposant la somme de l'équation précédente en deux termes : $n = N$ et $n \in [0; N-1]$, celle-ci peut s'écrire :

$$b_N p^N X_0(p) + \sum_{n=0}^{N-1} b_n p^n X_0(p) = E(p) \qquad \text{Eq. (161)}$$

En supposant alors que $b_N = 1$, ce qui permet, en outre, de normaliser les coefficients du filtre, $X_N(p)$ se définit par :

$$X_N(p) = -\sum_{n=0}^{N-1} b_n X_n(p) + E(p) \qquad \text{Eq. (162)}$$

Ainsi l'équation (159) devient :

$$\begin{aligned} S(p) &= a_N X_N(p) + \sum_{n=0}^{N-1} a_n X_n(p) \\ &= \sum_{n=0}^{N-1} (a_n - a_N b_n) X_n(p) + a_N E(p) \end{aligned} \qquad \text{Eq. (163)}$$

VI.2.2 Matrices d'état

A partir des équations (162) et (163), nous pouvons déduire les matrices de la représentation d'état qui interviennent dans l'équation (157). La matrice A (équation (164)) est appelée matrice d'état. Ses valeurs propres sont les pôles du filtre analogique car sous forme canonique, son polynôme caractéristique correspond au dénominateur de la fonction de transfert analogique du système. La matrice B (équation (165)) est appelée matrice de commande; C (équation (166)) matrice d'observation et D (équation (167)) matrice de liaison. Comme le coefficient de plus haut degré du dénominateur est fixé ($b_N = 1$), la causalité du système est assurée car le degré du dénominateur de la fonction de transfert est obligatoirement supérieur ou égal au

degré du numérateur. L'inconvénient est qu'il est impossible de réaliser des filtres RIF avec cette méthode.

$$A_{(N\times N)} = \begin{pmatrix} 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -b_0 & -b_1 & \cdots & -b_{N-2} & -b_{N-1} \end{pmatrix} \qquad \text{Eq. (164)}$$

$$B_{(N\times 1)} = \begin{pmatrix} 0 & 0 & \cdots & 1 \end{pmatrix}^T \qquad \text{Eq. (165)}$$

$$C_{(1\times N)} = \begin{pmatrix} a_0 - a_N b_0 & \cdots & a_{N-1} - a_N b_{N-1} \end{pmatrix} \qquad \text{Eq. (166)}$$

$$D_{(1\times 1)} = \begin{pmatrix} a_N \end{pmatrix} \qquad \text{Eq. (167)}$$

La fonction de transfert peut être obtenue facilement à partir de la représentation d'état grâce à la relation suivante :

$$H(p) = C(pI - A)^{-1} B + D \qquad \text{Eq. (168)}$$

Dans le domaine temporel, le vecteur d'état peut être obtenu en intégrant le système défini par l'équation (157) :

$$x(t) = e^{A(t-u)} x(0) + \int_0^t Be(u)du \qquad \text{Eq. (169)}$$

Finalement, la représentation dans l'espace d'état d'un filtre analogique peut être décrite par le schéma bloc suivant :

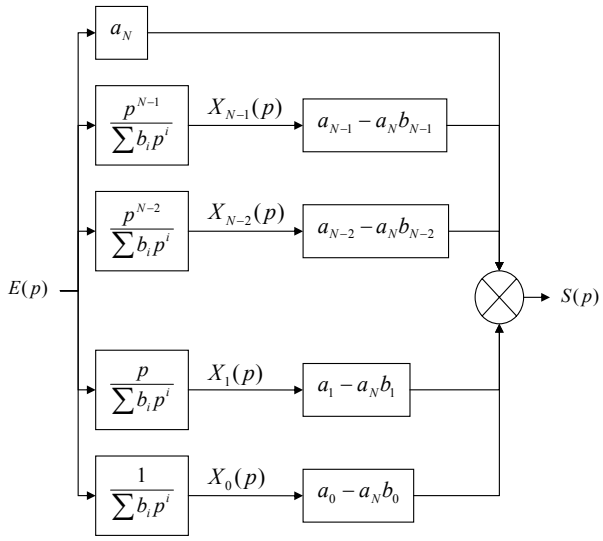


Figure 84 : Représentation dans l'espace d'état d'un filtre analogique

VI.3 Représentation d'un filtre numérique dans l'espace d'état

Pour obtenir un filtre numérique, il suffit alors de discrétiser l'espace d'état à l'aide d'une technique permettant d'approcher l'équation différentielle par une équation aux différences. Il existe beaucoup de techniques permettant de discrétiser l'espace d'état. Nous ne présenterons dans ce manuscrit que les méthodes les plus classiques ainsi que celles que nous avons utilisées dans nos travaux.

Par ailleurs comme pour chaque nouvel échantillon d'entrée, un échantillon est calculé en sortie, nous simplifierons les notations utilisées afin d'alléger les notations. Nous posons donc :

$$\begin{cases} te_n = ts_n = t_n \\ dte_n = dts_n = dt_n \\ ae_n = e_n \\ as_n = s_n \end{cases} \qquad \text{Eq. (170)}$$

VI.3.1 Méthodes standard

VI.3.1.1 Méthode d'Euler progressive

La méthode d'Euler progressive consiste à discrétiser le système à l'instant t_{n-1} et d'utiliser une approximation du 1er ordre de la dérivée :

$$\frac{x_n - x_{n-1}}{dt_n} = Ax_{n-1} + Be_{n-1} \qquad \text{Eq. (171)}$$

L'équation (157) s'écrit alors :

$$\begin{cases} x_n = (I + dt_n A)x_{n-1} + Bdt_n e_{n-1} \\ s_n = Cx_n + De_n \end{cases} \qquad \text{Eq. (172)}$$

VI.3.1.2 Méthode d'Euler rétrograde

La méthode d'Euler rétrograde consiste à discrétiser le système à l'instant t_n et d'utiliser une approximation du 1er ordre de la dérivée :

$$\frac{x_n - x_{n-1}}{dt_n} = Ax_n + Be_n \qquad \text{Eq. (173)}$$

L'équation (157) devient :

$$\begin{cases} x_n = (I - dt_n A)^{-1} x_{n-1} + dt_n B(I - dt_n A)^{-1} e_n \\ s_n = Cx_n + De_n \end{cases} \qquad \text{Eq. (174)}$$

VI.3.1.3 Méthode bilinéaire

D. Poulton et J. Oksman ont choisi dans leurs travaux [Poulton *et al.* 2001] la méthode bilinéaire aussi appelée Tustin ou encore méthode du trapèze car elle consiste à calculer une intégrale numériquement par une somme de trapèzes. En outre, elle permet d'approcher la dérivée d'un signal à un instant intermédiaire $t_{(n-1)/2} = (t_n + t_{n-1})/2$:

$$\frac{x_n - x_{n-1}}{dt_n} = A\frac{x_n + x_{n-1}}{2} + B\frac{e_n + e_{n-1}}{2} \qquad \text{Eq. (175)}$$

Ainsi, la représentation d'état peut se discrétiser sous la forme suivante :

$$\begin{cases} x_n = (I - \frac{dt_n}{2} A)^{-1} (I + \frac{dt_n}{2} A) x_{n-1} + \frac{dt_n}{2} (I - \frac{dt_n}{2} A)^{-1} B(e_n + e_{n-1}) \\ s_n = Cx_n + De_n \end{cases} \quad \text{Eq. (176)}$$

Ce schéma offre un certain nombre d'avantages : tout d'abord, il est d'ordre 2 et comme l'ont noté Poulton et Oksman, il est plus efficace que les méthodes d'Euler. Néanmoins, la complexité combinatoire est importante car pour un ordre *N* et pour chaque nouvel échantillon de sortie, une matrice *N*x*N* doit être inversée puis multipliée par une autre matrice *N*x*N*. C'est pour cette raison que nous suggérons d'utiliser d'autres schémas qui ne nécessitent pas une rétroaction imposant du même coup l'inversion d'une matrice.

VI.3.1.4 Résolution de l'équation différentielle

L. Fontaine et J. Ragot ont choisi de discrétiser la forme intégrale de l'espace d'état [Fontaine *et al.* 2001]. Leur seule approximation consiste simplement à remplacer le signal d'entrée analogique *e(t)* par un signal échantillonné bloqué :

$$\begin{cases} x_n = e^{Adt_n} x_{n-1} - A^{-1} (I - e^{Adt_n})^{-1} B e_{n-1} \\ s_n = Cx_n + De_n \end{cases} \quad \text{Eq. (177)}$$

La stabilité de cette méthode est assurée par le respect de la condition stabilité d'un filtre analogique dans l'espace d'état (que nous déterminons dans la section suivante). Bien qu'il n'y ait pas de comparaison entre différentes méthodes, les résultats présentés dans leur article sont de bonne qualité en terme de filtrage. Par ailleurs, ils suggèrent de décomposer un filtre d'ordre *N* en *N/2* filtres d'ordre 2 (où en *(N-1)/2* filtres d'ordre 2 et un filtre d'ordre 1 si *N* est impair) pour simplifier l'évaluation de e^{Adt_n}. Nous utiliserons cette idée pour comparer les complexités combinatoires des différentes méthodes et proposer des architectures matérielles adéquates.

VI.3.2 Conditions de stabilité et flot de conception

VI.3.2.1 Conditions de stabilité

VI.3.2.1.1 Définition

Un filtre est défini comme stable si la sortie retourne à un état d'équilibre après que le système ait été perturbé par un signal d'entrée de durée finie.

Une condition nécessaire et suffisante pour assurer la stabilité d'un filtre analogique est que les pôles de la fonction de transfert soient à partie réelle négative. Dans la représentation d'état définie précédemment, les pôles du filtre correspondent aux racines du polynôme caractéristique :

$$det(\lambda I - A) = \lambda^N + b_{N-1}\lambda^{N-1} + b_1\lambda + b_0 \qquad \text{Eq. (178)}$$

qui sont précisément les valeurs propres de la matrice d'état.

VI.3.2.1.2 Stabilité de la représentation d'état à temps continu

Supposons que le signal d'entrée devienne constant à partir d'un temps t_1. Dans ce cas, la solution du système pour tout temps $t \geq t_1$ s'écrit :

$$\begin{cases} x(t) = e^{A(t-t_1)}x(t_1) + \int_{t_1}^{t} e^{A(t-u)}Be(u)du \\ \quad\quad = e^{A(t-t_1)}x(t_1) + A^{-1}\left(e^{A(t-t_1)} - I\right)Be(t_1) \\ s(t) = Cx(t) + De(t) \end{cases} \qquad \text{Eq. (179)}$$

Puisque les valeurs propres de la matrice A sont à partie réelle négative, nous pouvons calculer la valeur limite de la sortie du filtre :

$$\begin{aligned} \lim_{t\to+\infty} s(t) &= \left(D - CA^{-1}B\right)e(t_1) \\ &= \left(a_N + \frac{a_0 - a_N b_0}{b_0}\right)e(t_1) \\ &= \frac{a_0}{b_0}e(t_1) \end{aligned} \qquad \text{Eq. (180)}$$

Nous observons que cette limite ne dépend pas de l'état du système mais uniquement d'une valeur d'entrée constante. Par conséquent, nous obtenons bien la preuve que le système est stable si les valeurs propres de la matrice A sont à partie réelle négative.

VI.3.2.1.3 Stabilité de la représentation d'état à temps discret

Lors du passage théorique à temps discret, une matrice e^{At} résulte de l'équation différentielle sans second membre. La condition de stabilité qui dépendait de la partie réelle des valeurs propres de la matrice A devient alors fonction du module des valeurs propres de e^{At} : celles-ci doivent en effet être de module inférieur à 1 c'est-à-dire être incluses dans le cercle unité. Or, dans un cas pratique cette condition doit être respectée quel que soit le schéma utilisé pour approcher e^{At} lors de la discrétisation de l'espace d'état.

Ainsi, en généralisant la représentation d'état à temps discret par l'équation suivante :

$$x_n = \Phi_n x_{n-1} + \Gamma_n e_{n-1} \qquad \text{Eq. (181)}$$

les valeurs propres de la matrice Φ_n qui approche e^{Adt_n} doivent appartenir au cercle unité. L'étude de la stabilité ne dépend plus que de la fonction T_n qui projette les valeurs propres λ de A vers celles de Φ_n notées $\mu_n = T_n(\lambda)$. Par exemple, pour la méthode d'Euler rétrograde, il est connu que l'image du demi plan gauche par la transformation définie par $T_n(\lambda) = (1-dt_n\,\lambda)^{-1}$ est un cercle de centre ½ et de rayon ½. Ainsi nous pouvons en déduire par avance que tout filtre analogique stable sera numériquement stable. En revanche pour la méthode d'Euler progressive, la transformation est $T_n(\lambda) = 1+dt_n\,\lambda$; ainsi toute valeur propre à partie réelle négative est projetée dans une région dont les parties réelles sont inférieures à 1, incluant le cercle unité; la stabilité n'est donc pas assurée. Enfin avec la méthode bilinéaire, la fonction vaut $T_n(\lambda) = \dfrac{1+dt_n\lambda/2}{1-dt_n\lambda/2}$ projetant le demi-plan gauche dans le cercle unité permettant de garantir lors de la conception une stabilité inconditionnelle.

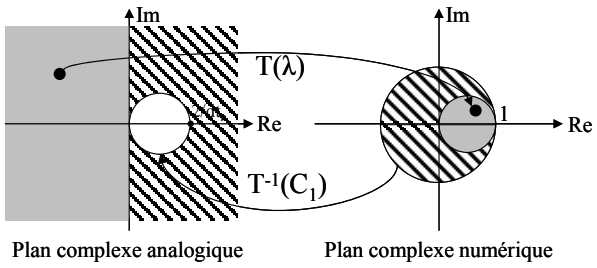


Figure 85 : Transformation du demi-plan gauche avec la méthode d'Euler rétrograde

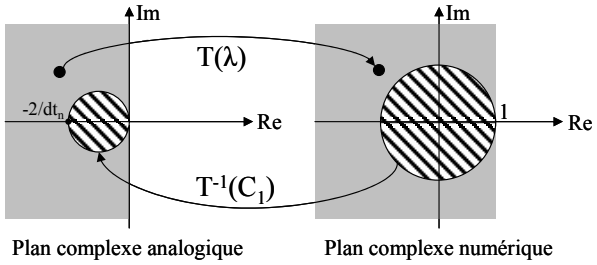


Figure 86 : Transformation du demi-plan gauche pour la méthode d'Euler progressive

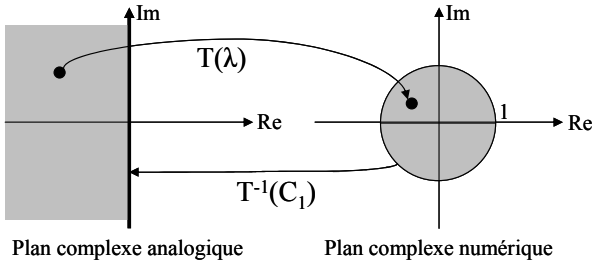


Figure 87 : Transformation du demi-plan gauche pour la méthode bilinéaire

VI.3.2.2 Flot de conception

Lors de la conception, une discussion doit donc être réalisée sur le choix de la méthode selon que l'on veut :

- que tout filtre analogique stable devienne un filtre numérique stable,
- partir d'une classe particulière de filtres analogiques dont les pôles appartiennent à une région telle que leur projection soit inclue dans le cercle unité.

Les méthodes d'Euler rétrograde et bilinéaire respectent la première contrainte tandis que la méthode d'Euler progressive par exemple ne permet de concevoir des filtres numériques stables qu'à partir de filtres dont les pôles sont inclus dans un cercle de centre $-1/dt_n$ et de rayon $1/dt_n$ comme le représente la Figure 86.

Cette discussion conduit à la définition d'un flot de conception pour généraliser l'implémentation d'un filtre numérique stable à partir d'une méthode donnée. Le flot est représenté sur la Figure 88. Il y a deux entrées : le schéma numérique et le filtre analogique, c'est-à-dire ses pôles (obtenus à partir du gabarit). Pour un intervalle de temps donné, une transformation T_n est calculée ainsi que la région R_n du plan complexe analogique définie comme l'image inverse du cercle unité C_1 : $R_n = T_n^{-1}(C_1)$. Si le demi plan gauche est inclus dans R_n alors le schéma numérique est une méthode stable pour tout filtre analogique. Dans le cas contraire, si les pôles du filtre analogique appartiennent à R_n, alors le schéma est stable pour ce filtre particulier. Si la condition n'est pas respectée, une solution possible est alors de réduire la valeur des intervalles de temps puisqu'ils agissent sur la région R_n comme facteur d'homothétie.

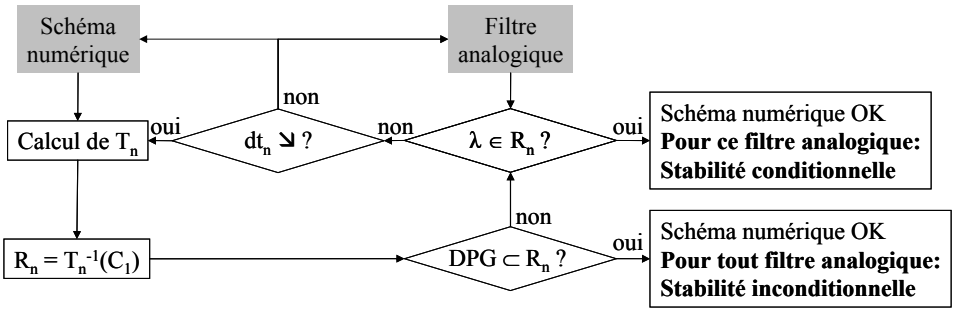


Figure 88 : Flot de conception d'un filtre numérique stable

S'il n'est pas possible de modifier la valeur de l'intervalle, la solution consiste alors à soit modifier le filtre analogique en vérifiant que les pôles sont inclus dans R_n pour assurer la stabilité du filtre numérique soit de changer le schéma numérique.

VI.3.3 Schémas numériques généralisés

VI.3.3.1 Schémas explicites et implicites

Lors de la discrétisation de la représentation d'état défini par l'équation (157), le membre de gauche, c'est-à-dire la dérivée du vecteur d'état, est toujours remplacé par $(x_n-x_{n-1})/dt_n$. Si dans la partie de droite, le vecteur d'état et l'entrée sont approchées par des échantillons aux instants

précédant t_n i.e. t_{n-1}, t_{n-2},… alors le schéma numérique est qualifié d'*explicite*. Nous avons déjà présenté par exemple la méthode d'Euler progressive qui est un schéma explicite (cf. équation (172)). Si le temps t_n est utilisé, le schéma est défini comme *semi-implicite*; la méthode bilinéaire en est un exemple (équation (176)). Enfin si des temps postérieurs à t_n sont utilisés comme t_{n+1}, le schéma est dit *implicite*. De telles méthodes ne peuvent être utilisées car elles nécessitent de connaître non seulement des entrées qui ne sont pas encore échantillonnées mais aussi une estimation des futurs états du système.

VI.3.3.2 Schéma explicite Runge-Kutta 4

Il existe de nombreux schémas numériques dans la littérature [Gear 1971] [Crouzeix *et al.* 1992], avec parmi eux, les méthodes explicites de Runge-Kutta. Nous proposons d'utiliser un schéma de Runge-Kutta d'ordre 4 pour discrétiser la représentation d'état. L'équation (157) devient alors :

$$\begin{cases} x_n = \left(I + dt_n A + \frac{1}{2} dt_n^2 A^2 + \frac{1}{6} dt_n^3 A^3 + \frac{1}{24} dt_n^4 A^4 \right) x_{n-1} \\ \quad + dt_n \left(\frac{1}{2} I + \frac{1}{3} dt_n A + \frac{1}{8} dt_n^2 A^2 + \frac{1}{24} dt_n^3 A^3 \right) B e_{n-1} \\ \quad + dt_n \left(\frac{1}{2} I + \frac{1}{6} dt_n A + \frac{1}{24} dt_n^2 A^2 \right) B e_n \\ s_n = C x_n + D e_n \end{cases} \qquad \text{Eq. (182)}$$

Nous pouvons noter que la matrice Φ_n est une approximation obtenue par le développement de Taylor à l'ordre 4 de $e^{A dt_n}$.

Conformément au flot de conception, nous étudions la transformation du plan complexe analogique vers le plan complexe numérique. Pour ce schéma, la transformation est donnée par l'équation suivante :

$$T(\lambda) = 1 + dt_n \lambda + \frac{1}{2} dt_n^2 \lambda^2 + \frac{1}{6} dt_n^3 \lambda^3 + \frac{1}{24} dt_n^4 \lambda^4 \qquad \text{Eq. (183)}$$

La transformation inverse du cercle unité $R_n = T_n^{-1}(C_1)$ est représentée sur la Figure 89.

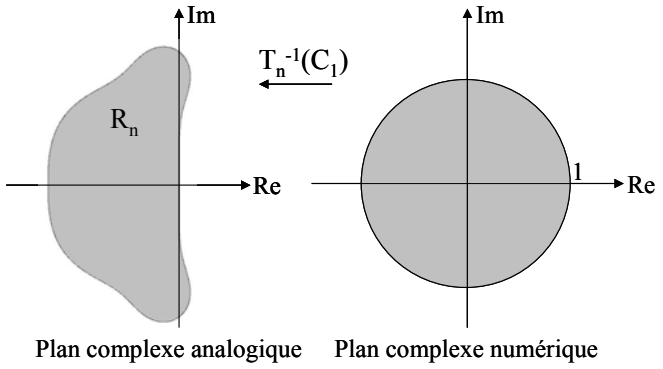


Figure 89 : Transformation inverse du cercle unité pour la méthode Runge-Kutta d'ordre 4

Nous pouvons en déduire que la méthode n'est pas inconditionnellement stable puisque le demi plan gauche n'est pas inclus dans R_n. Ce résultat était en réalité prévisible car une méthode explicite ne peut pas être inconditionnellement stable. En effet, dans ce cas, la transformation T_n est une fonction polynomiale et aucune fonction polynomiale ne peut projeter le demi-plan gauche dans un espace borné du plan complexe numérique, a fortiori dans le cercle unité.

VI.3.3.3 Schéma semi-implicite de Runge-Kutta 3

Parmi les méthodes de Runge-Kutta, il existe un cas particulier pour lequel le schéma devient semi-implicite : il s'agit de Runge-Kutta d'ordre 3 à 2 étages, noté RK23. La représentation d'état s'écrit alors :

$$\begin{cases} x_n = \left(I + dt_n A\right)^{-1}\left(I + \frac{2}{3}dt_n A\right)^{-1} y_n \\ y_n = \left(I - \frac{2}{3}dt_n A - \frac{1}{2}dt_n^2 A^2\right)x_{n-1} + \frac{dt_n}{2}\left(I - dt_n A\right)Be_{n-1} + \frac{dt_n}{2}\left(I - 3dt_n A\right)Be_n \\ s_n = Cx_n + De_n \end{cases} \quad \text{Eq. (184)}$$

Nous en déduisons la transformation du plan complexe analogique :

$$T_n(\lambda) = \frac{1 - \frac{2}{3}dt_n\lambda - \frac{1}{2}dt_n^2\lambda^2}{\left(1 + dt_n\lambda\right)\left(1 + \frac{2}{3}dt_n\lambda\right)} \quad \text{Eq. (185)}$$

Comme toute méthode semi-implicite, le schéma RK23 est inconditionnellement stable. En effet, le demi-plan gauche est projeté dans un espace inclus dans le cercle unité; réciproquement le demi-plan gauche est inclus dans la transformation inverse du cercle unité, comme le montre la Figure 90.

En revanche, comme pour la méthode bilinéaire, le schéma RK23 nécessite une inversion de matrice, très coûteuse pour une implémentation matérielle.

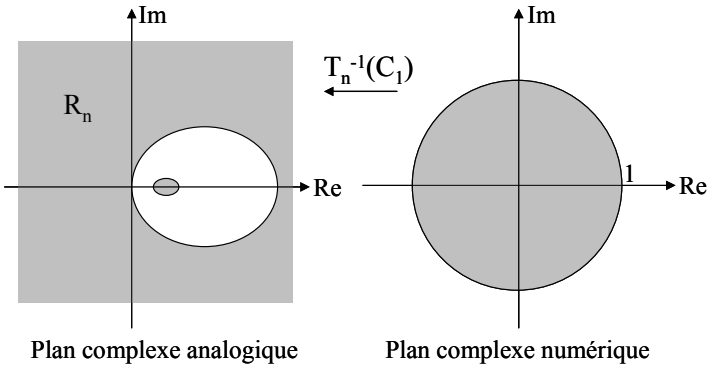


Figure 90 : Transformation inverse du cercle unité pour la méthode de Runge-Kutta d'ordre 3 à 2 étages

VI.4 Réponse en fréquence du filtre numérique

Une des difficultés dans la conception de filtre numérique RII est de respecter la réponse fréquentielle spécifiée dans le cahier des charges, à l'origine du filtre analogique de départ. En effet, lors de la discrétisation, l'axe imaginaire sur lequel sont projetés les pôles et les zéros analogiques pour passer de la transformée de Laplace à la transformée de Fourier ($p = j2\pi f$), est théoriquement transformé en cercle unité. Dans le cadre d'un échantillonnage régulier de fréquence $F_e = 1/T_e$, les pôles et zéros numériques sont alors projetés sur le cercle pour obtenir la réponse en fréquence du filtre numérique. Comme le point (1,0) correspond aux fréquences $f = kF_e$ pour tout entier relatif k, la réponse en fréquence du filtre analogique est simplement périodisée conformément à la théorie de l'échantillonnage classique appliquée à sa réponse impulsionnelle.

Dans le cas qui nous intéresse, deux points vont à l'encontre de ce qui vient d'être affirmé. Tout d'abord, l'utilisation d'un schéma numérique ne transforme pas nécessairement l'axe imaginaire en cercle unité ce qui implique une déformation de la réponse en fréquence analogique. Toutefois, cela est également vrai avec un signal échantillonné uniformément. D'autre part, comme

l'échantillonnage est non uniforme, il y a autant de manières de parcourir la transformée de l'axe imaginaire, qu'il y a d'intervalles de temps (c'est-à-dire de « fréquences d'échantillonnage »). Il n'est donc pas possible de déterminer la réponse en fréquence du filtre numérique. Ce fait est également avéré par l'impossibilité de retrouver la réponse impulsionnelle numérique à partir de la représentation d'état. En effet, étant donné que l'équation de récurrence entre la sortie et l'entrée dépend des intervalles de temps, le système est rendu non linéaire et non invariant dans le temps empêchant l'utilisation d'une transformée en Z. La réponse en fréquence est donc inconnue. Toutefois, deux hypothèses peuvent être formulées pour lever l'indétermination.

1. Pour des fréquences proches de $f = 0$, la transformation de l'axe imaginaire tend vers le cercle unité quelle que soit la méthode utilisée. La déformation de la réponse en fréquence est alors suffisamment faible pour approcher la réponse en fréquence du filtre numérique par celle du filtre analogique dans cette zone. Sur la Figure 91 sont représentées les projections de l'axe imaginaire par la transformation de chaque schéma numérique. Nous pouvons remarquer que cette hypothèse est justifiée à des degrés divers selon la méthode. Ainsi par exemple, les méthodes d'Euler divergent rapidement les rendant inutiles du point de vue spectral.

2. La déformation augmente vers les hautes fréquences. Dans le cas régulier, la fréquence d'échantillonnage est la limite supérieure de la déformation du fait de la périodisation du spectre. Or, plus la fréquence d'échantillonnage est élevée devant la bande utile du filtre (bande passante, bande transition et le début de la bande atténuée majorée par F_{MAX}), plus la déformation s'opère sur les hautes fréquences, comme le montre la Figure 92. Si la fréquence d'échantillonnage tend vers l'infini, la déformation est nulle : la réponse en fréquence du filtre numérique tend vers celle du filtre analogique puisque la réponse impulsionnelle du filtre numérique tend vers celle du filtre analogique. Dans le cadre d'un échantillonnage non uniforme, nous pouvons donc supposer que plus l'inverse de l'ensemble des intervalles de temps est grand devant la bande utile du filtre, plus la déformation, relative à la bande utile, sera faible. Or, nous savons que l'échantillonnage par traversée de niveaux suréchantillonne systématiquement le signal quand il y a des variations importantes. Ainsi, lors des périodes inactives, aucun point n'est traité ; puis lorsqu'une variation apparaît, un point avec un intervalle de temps important est échantillonné. La réponse en fréquence est alors déformée dans la bande utile du filtre; la valeur de sortie risque donc d'être *fausse* mais cela n'a en réalité pas d'importance puisque ce point sert de

réamorçage, c'est-à-dire de régime transitoire au traitement de la zone active. Il faut juste s'assurer que l'intervalle de temps ne déstabilise pas le filtre si la stabilité est conditionnelle. Les points suivants, appartenant à la zone active du signal, ont des intervalles de temps plus petits du fait du suréchantillonnage ; il n'y a donc pas de déformation de la réponse en fréquence du filtre jusqu'à la période inactive suivante. Il faut s'assurer, par conséquent, que l'ensemble des inverses des intervalles de temps (excepté le plus grand) soit grand devant la fréquence maximale de la bande utile.

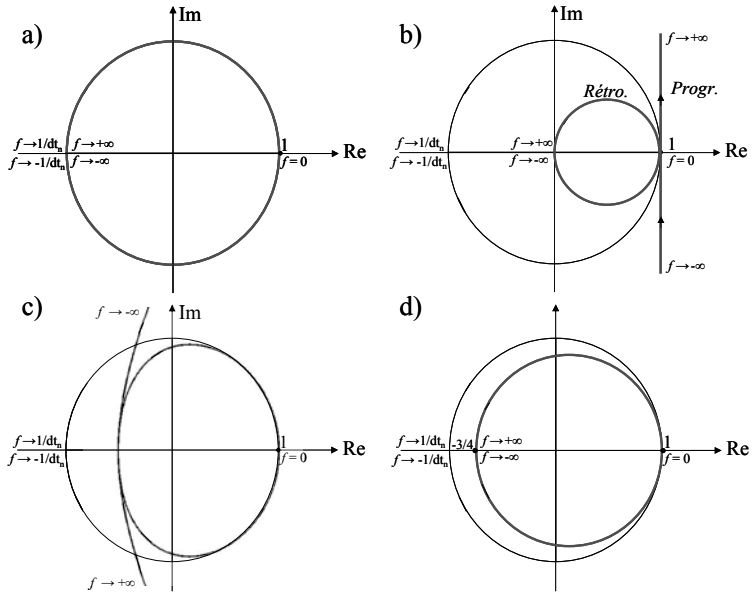


Figure 91 : Projection de l'axe imaginaire du plan complexe analogique par les transformations des méthodes bilinéaire (a), d'Euler rétrograde et progressive (b), Runge-Kutta d'ordre 4 (c) et RK23 (d)

Parallèlement à ce qui vient d'être vu, la méthode bilinéaire possède un avantage connu des concepteurs de filtre numérique RII : la déformation de la réponse en fréquence est réelle, c'est-à-dire que le passage des fréquences analogiques aux fréquences numériques est une fonction qu'il est possible d'inverser. Ainsi en prédéformant la réponse en fréquence du filtre analogique, il est aisé d'obtenir le filtre numérique ayant la réponse en fréquence désirée. Cependant dans le cadre de l'échantillonnage non uniforme, cet avantage est inapplicable pour deux raisons. La première est que la réponse en fréquence numérique est inconnue. La déformation est alors également inconnue ; il n'est donc pas possible de faire une rétroaction. La seconde est qu'en admettant que cela puisse être le cas, il faudrait alors calculer un nouveau filtre analogique en fonction de la nouvelle valeur de l'intervalle de temps en cours. Les matrices A et C de la représentation d'état

qui étaient jusqu'à présent des constantes, deviendraient des variables A_n et C_n. La matrice de passage Φ_n entre x_n et x_{n-1} serait alors $(I - \frac{dt_n}{2} A_n)^{-1}(I + \frac{dt_n}{2} A_n)$, compliquant considérablement tout implémentation matérielle.

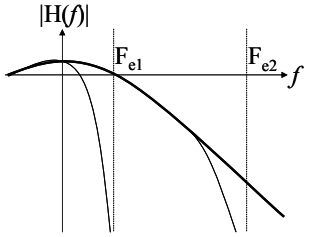


Figure 92 : Exemple de réponses en fréquence déformées par la méthode bilinéaire pour deux valeurs de fréquence d'échantillonnage (les périodisations ne sont affichées pour plus de clarté)

VI.5 Exemples de filtrage

Nous proposons d'illustrer le filtrage RII de signaux échantillonnés non uniformément pour les différents schémas numériques discrétisant l'espace d'état. Dans la mesure où nous voulons mettre en évidence l'influence des pôles, nous nous limiterons à ne présenter que les filtrages passe-bas – les autres types de filtrage se caractérisant par des zéros au numérateur de la fonction de transfert. Comme pour le filtrage FIR, le signal d'entrée comporte deux composantes : une fréquence fondamentale de 1Hz et son harmonique 5 à 5Hz. Le signal est échantillonné sur 15 niveaux répartis régulièrement sur sa dynamique.

Comme un filtre RII est conçu à partir d'un filtre analogique, nous définissons un gabarit en module : fréquence de coupure $f_c = 2Hz$, fréquence atténuée $f_a = 5Hz$, oscillation autorisée dans la bande passante 1dB, gain de la bande atténuée -40dB. Nous réalisons deux filtres à l'aide de deux fonctions d'approximation différentes : un filtre de Butterworth d'ordre 6 et un filtre de Chebyshev d'ordre 4. Les pôles sont représentés dans le plan complexe analogique sur la Figure 93.

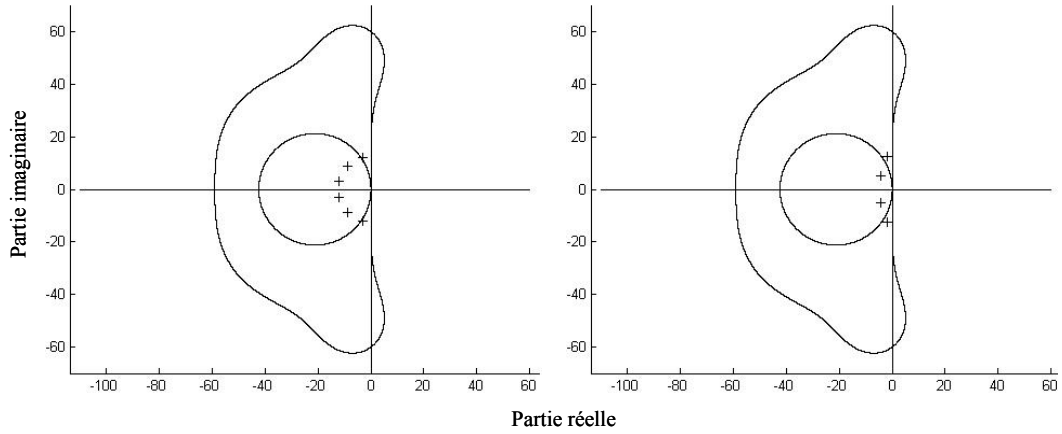


Figure 93 : Représentation dans le plan complexe analogique des pôles des filtres de Butterworth d'ordre 6 (à gauche) et de Chebyshev d'ordre 4 (à droite)

Sur la Figure 93, sont également représentés les plus petits sous-espaces du plan complexe définis par les transformations inverses du cercle unité pour les deux schémas explicites (obtenus en utilisant l'intervalle de temps le plus grand), c'est-à-dire les méthodes d'Euler et Runge-Kutta d'ordre 4. Nous pouvons noter que pour les deux filtres, une paire de pôles se trouve à l'extérieur du sous-espace de la méthode d'Euler, impliquant une instabilité.

Les résultats du filtrage sont présentés respectivement sur les Figure 94 et Figure 95 pour les filtres de Butterworth et de Chebyshev. La méthode d'Euler mise à part, l'ensemble des signaux convergent vers un signal sinusoïdal ayant une composante continue. Il est à noter que la fonction d'approximation de Chebyshev qui autorise une ondulation dans la bande passante contribue à atténuer la valeur moyenne et l'amplitude crête du signal de sortie. En regardant de plus près les différences entre les signaux de sortie, nous pouvons observer que les schémas bilinéaire, de Runge-Kutta d'ordre 4 et la méthode de résolution directe de l'équation différentielle obtiennent des résultats similaires. En revanche, les méthodes d'Euler rétrograde et de Runge-Kutta semi-implicite d'ordre 3 donnent des résultats différents qui sont liés à la modification de la réponse en fréquence. Il s'agit là du constat habituel, notamment pour la méthode d'Euler rétrograde, qui atténue, dans l'ensemble, la courbe de gain. Enfin pour la méthode d'Euler progressive, la présence de pôles instables provoque une divergence du signal de sortie. Cependant, nous pouvons observer, avec le filtre de Butterworth, que l'instabilité n'induit qu'une surtension du signal. Nous proposons d'étudier ce phénomène, à travers le schéma de Runge-Kutta d'ordre 4 car il possède un sous-espace plus grand que la méthode d'Euler. De ce fait, il est plus difficile à déstabiliser ce qui le rend plus intéressant à utiliser.

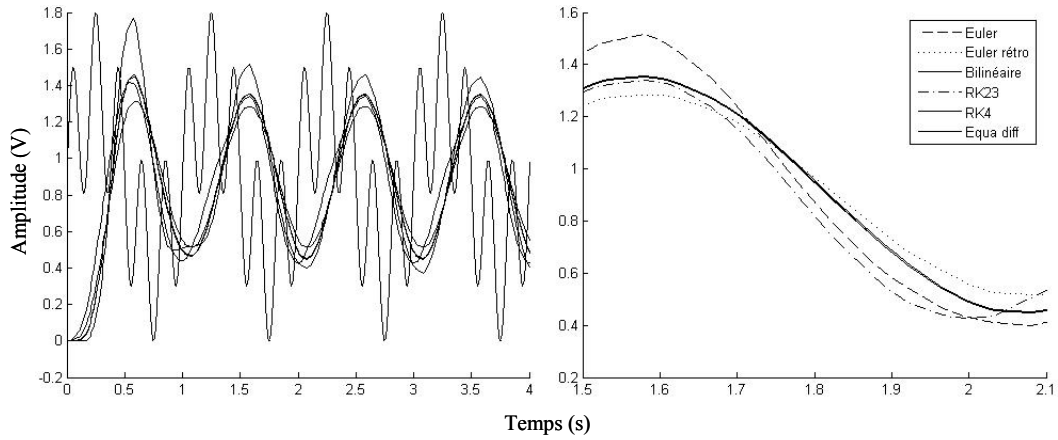


Figure 94 : Filtre de Butterworth d'ordre 6 (plan large et zoom)

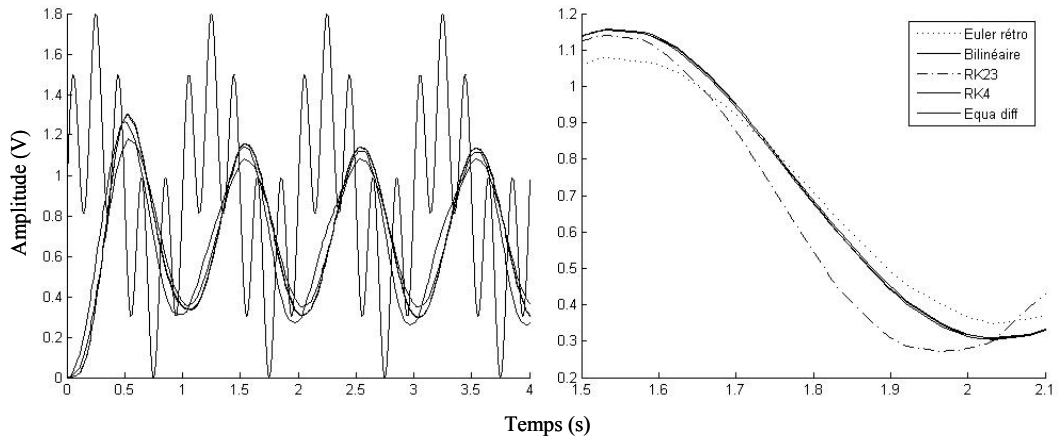


Figure 95 : Filtre de Chebyshev d'ordre 4 (plan large et zoom)

Nous proposons donc de réaliser deux nouveaux filtres pour traiter un signal sinusoïdal pur à l'aide de la méthode de Runge-Kutta d'ordre 4. Afin d'observer un signal de sortie semblable au signal d'entrée, nous élevons la fréquence de coupure à $f_c = 40Hz$. Puis, pour étudier l'influence de la position des pôles, nous concevons deux filtres de même ordre i.e. d'ordre 10. Compte tenu de la fréquence de coupure et de l'ordre, les deux filtres sont instables pour cette entrée donnée, c'est-à-dire pour un intervalle de temps maximum donné. Celui-ci induit un sous-espace du plan complexe analogique hors duquel sont exclus certains pôles. Pourtant, les signaux de sortie ne sont pas divergents comme le montrent les Figure 96 et Figure 97.

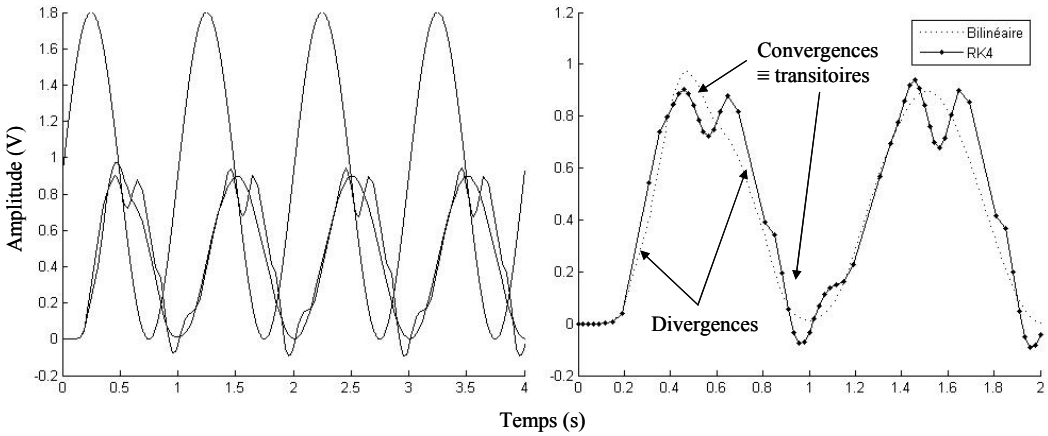


Figure 96 : Filtre de Butterworth d'ordre 10 déstabilisant la méthode RK4 (plan large et zoom)

En réalité, chaque intervalle de temps induit un sous-espace du plan complexe particulier. Plus l'intervalle est grand, plus le sous-espace est petit. Ainsi, pour un ensemble d'intervalles de temps donné, seule une partie d'entre eux induisent des sous-espaces excluant des pôles. Lorsque l'un de ces intervalles est utilisé, le filtre devient instable et le signal diverge. Mais si, lors des calculs suivants, les intervalles de temps sont suffisamment petits, le filtre redevient stable. Le signal de sortie converge à nouveau ; un transitoire apparaît jusqu'à la prochaine divergence. Si plusieurs intervalles de temps consécutifs déstabilisent le filtre, le signal ne peut plus converger ; la divergence est définitive.

Il est important de noter que la fonction d'approximation choisie lors de la conception n'influe pas sur ce phénomène. En effet, les pôles issus du polynôme de Butterworth se situent sur un cercle dépendant de la fréquence de coupure quel que soit l'ordre du filtre. Changer l'ordre du filtre n'a donc aucune influence. Les pôles issus du polynôme de Chebyshev sont situés sur une ellipse qui dépend également de la fréquence de coupure mais aussi de l'ondulation autorisée dans la bande passante. Cependant, plus l'ordre diminue, plus le petit rayon augmente. La paire de pôles qui se trouvait à l'extérieur du sous-espace reste à l'extérieur quel que soit l'ordre (sauf cas particulier de l'ordre 4). Ainsi, il ne sert à rien de privilégier une fonction particulière par rapport à une autre vis-à-vis de la stabilité ; seul le gabarit permet, pour un signal donné, de conclure sur la stabilité.

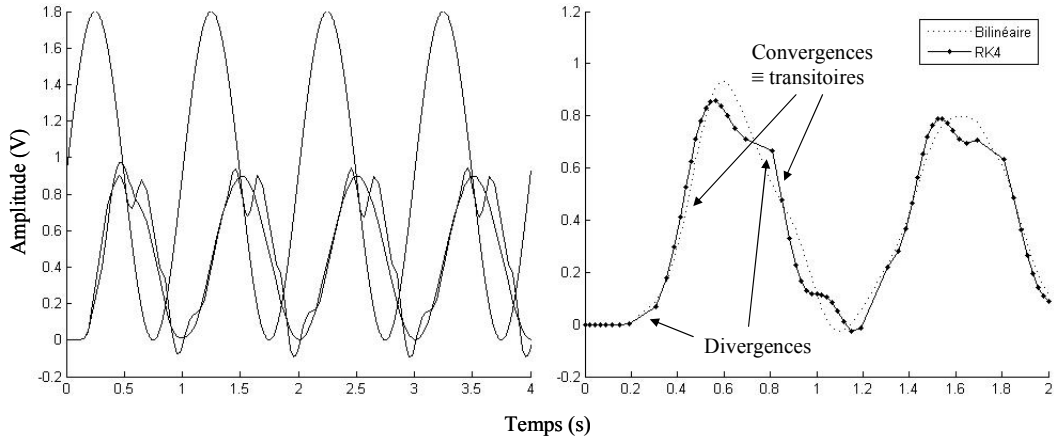


Figure 97 : Filtre de Chebyshev d'ordre 10 déstabilisant la méthode RK4 (plan large et zoom)

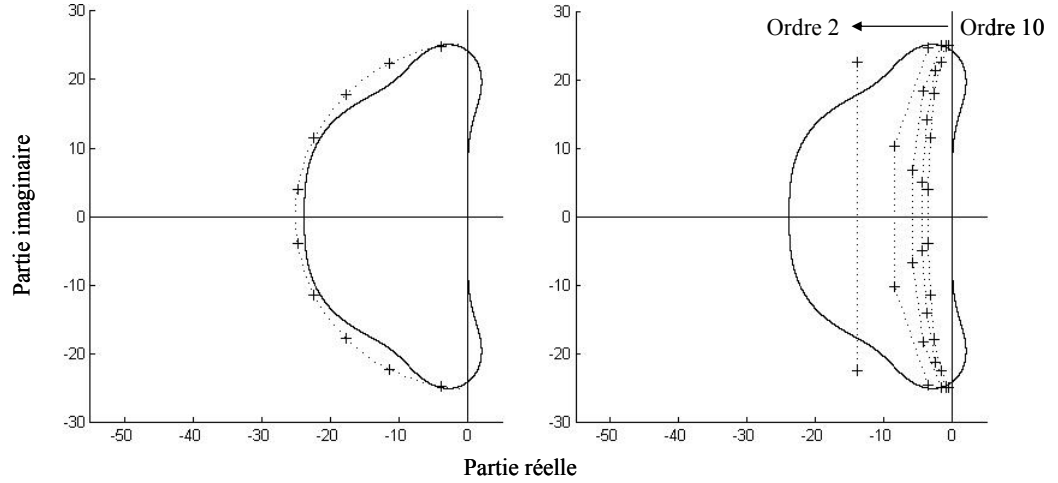


Figure 98 : Représentation dans le plan complexe analogique des pôles des filtres de Butterworth d'ordre 10 (à gauche) et de Chebyshev pour les ordres 2 à 10 (à droite)

VI.6 Architectures matérielles

Compte tenu des résultats de simulation, nous n'étudierons pas les méthodes d'Euler car dans le cas rétrograde (inconditionnellement stable), elle est moins efficace que la méthode bilinéaire ou RK23 et dans le cas progressif (conditionnellement stable), elle est également moins efficace que RK4.

VI.6.1 Analyse des complexités combinatoires des schémas numériques

VI.6.1.1 Echantillonnage uniforme

Dans le cas de l'échantillonnage uniforme, un filtre RII est défini par son équation aux différences. Bien que leurs valeurs dépendent du schéma numérique utilisé, les coefficients de l'équation sont constants quelle que soit la méthode. La complexité est donc fixée et donnée dans le tableau suivant :

Schéma	Additions	Multiplications	Décalages	Exp.	Mémoires
TZ	2*N*	2*N*+1	/	/	2*N* (forme directe 1) *N* (forme directe 2)

Tableau 5 : Complexité combinatoire d'un filtre RII échantillonné régulièrement

VI.6.1.2 Echantillonnage non uniforme

Nous avons vu précédemment que les schémas numériques implicites ou semi-implicites nécessitent une inversion de matrice lors du calcul de l'état courant x_n. En relâchant la contrainte sur la stabilité du système, nous avons établi que les schémas explicites se préservent d'une quelconque inversion, coûteuse en ressource matérielle tandis que la technique de résolution de l'équation différentielle, inconditionnellement stable, implique le calcul d'une exponentielle de matrice.

Nous avons voulu comparer les complexités combinatoires de chaque schéma pour obtenir un critère supplémentaire dans le choix de la méthode à utiliser pour discrétiser la représentation d'état. Pour les schémas implicites, l'inversion de matrice empêche toute implémentation directe d'un filtre d'ordre *N*. Le filtre doit donc être décomposé soit en structures simples du 2ème ordre en

regroupant par deux les paires de pôles complexes conjugués, soit en structures simples du 1er ordre pour chaque pôle réel supplémentaire. D'une manière générale, nous considérerons qu'un filtre d'ordre *N* sera décomposé en *N/2* filtre du 2ème ordre si *N* est pair ou *(N-1)/2* filtres du 2ème ordre et 1 filtre du 1er ordre si *N* est impair. En effet, il est rare qu'un filtre analogique obtenu à partir d'une fonction d'approximation ait plusieurs pôles réels – avec les polynômes de Butterworth, par exemple, les pôles se situent sur un cercle de centre 0 dont le rayon est fonction de la fréquence de coupure. Avec la résolution de l'équation différentielle, le filtre devra être décomposé de la même manière pour utiliser la méthode de Cayley-Hamilton dans le calcul de l'exponentielle de la matrice *A*. En revanche, avec le schéma explicite de Runge-Kutta RK4, un calcul direct du filtre à l'ordre *N* pourra être envisagé.

VI.6.1.2.1 Structures simples du 1er et 2ème ordre

Pour les méthodes bilinéaire, RK23 et résolution de l'équation différentielle, nous avons donc calculé séparément le nombre d'opération nécessaire pour un filtre d'ordre *N* décomposé en structures du 1er et du 2ème ordre. Les résultats sont présentés sur le Tableau 6.

Schéma	Additions	Multiplications	Décalages	Exp.	Mémoires
Bilinéaire	$10\left\lfloor \frac{N}{2} \right\rfloor + 5N \bmod 2$	$18\left\lfloor \frac{N}{2} \right\rfloor + 7N \bmod 2$	$2\left\lfloor \frac{N}{2} \right\rfloor + 2N \bmod 2$	/	$3\left\lfloor \frac{N}{2} \right\rfloor + 2N \bmod 2$
RK23	$26\left\lfloor \frac{N}{2} \right\rfloor + 8N \bmod 2$	$44\left\lfloor \frac{N}{2} \right\rfloor + 9N \bmod 2$	$\left\lfloor \frac{N}{2} \right\rfloor + N \bmod 2$	/	$3\left\lfloor \frac{N}{2} \right\rfloor + 2N \bmod 2$
Résol. ED	$13\left\lfloor \frac{N}{2} \right\rfloor + 3N \bmod 2$	$20\left\lfloor \frac{N}{2} \right\rfloor + 6N \bmod 2$	/	*N*	$3\left\lfloor \frac{N}{2} \right\rfloor + 2N \bmod 2$

Tableau 6 : Complexités combinatoires des schémas décomposables en une structure simple

VI.6.1.2.2 Schéma de Runge-Kutta d'ordre 4

La représentation d'état discrétisée par le schéma de Runge-Kutta d'ordre 4 définie par l'équation (182) peut se calculer itérativement en 4 étapes grâce à l'utilisation de 4 variables intermédiaires, chacune de longueur *N*, $k_1=(k_{10}, k_{11}, \ldots k_{1,N-1})^T$, $k_2=(k_{20}, k_{21}, \ldots k_{2,N-1})^T$, $k_3=(k_{30}, k_{31}, \ldots$

$k_{3,N-1)}^T$, $k_4=(k_{40}, k_{41}, \ldots k_{4,N-1)}^T$. L'équation numérique récursive est alors simplifiée par le système suivant :

$$\begin{cases} k_1 = Ax_{n-1} + Be_{n-1} \\ k_2 = Ax_{n-1} + \dfrac{dt_n}{2}k_1 + B\dfrac{e_{n-1}+e_n}{2} \\ k_3 = Ax_{n-1} + \dfrac{dt_n}{2}k_2 + B\dfrac{e_{n-1}+e_n}{2} \\ k_4 = Ax_{n-1} + dt_n k_3 + Be_n \end{cases} \quad \text{Eq. (186)}$$

$$\begin{cases} x_n = x_{n-1} + \dfrac{dt_n}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\ s_n = Cx_n + De_n \end{cases} \quad \text{Eq. (187)}$$

Nous pouvons ainsi comptabiliser simplement le nombre d'opérations en fonction de l'ordre du filtre en regroupant les $(N-1)^{ères}$ lignes des vecteurs k_1, k_2, k_3 et k_4 :

$$\begin{cases} k_{1,i} = x_{i+1,n-1} \\ k_{2,i} = x_{i+1,n-1} + \dfrac{dt_n}{2}k_{1,i+1} \\ k_{3,i} = x_{i+1,n-1} + \dfrac{dt_n}{2}k_{2,i+1} \\ k_{4,i} = x_{i+1,n-1} + \dfrac{dt_n}{2}k_{3,i+1} \end{cases} \quad \text{Eq. (188)}$$

Chaque ligne d'une variable k est calculée en fonction de la ligne suivante du vecteur d'état à l'instant précédent et de la ligne suivante du vecteur k précédent (sauf pour k_1 qui n'en a pas). La dernière ligne de chaque vecteur k est parallèlement calculée de la manière suivante :

$$\begin{cases} k_{1,N-1} = -\sum_{j=0}^{N-1} b_j x_{j,n-1} + e_{n-1} \\ k_{2,N-1} = -\sum_{j=0}^{N-1} b_j x_{j,n-1} - \dfrac{dt_n}{2}\sum_{j=0}^{N-1} b_j k_{1,j} + \dfrac{e_{n-1}+e_n}{2} \\ k_{3,N-1} = -\sum_{j=0}^{N-1} b_j x_{j,n-1} - \dfrac{dt_n}{2}\sum_{j=0}^{N-1} b_j k_{2,j} + \dfrac{e_{n-1}+e_n}{2} \\ k_{4,N-1} = -\sum_{j=0}^{N-1} b_j x_{j,n-1} - dt_n\sum_{j=0}^{N-1} b_j k_{3,j} + e_n \end{cases} \quad \text{Eq. (189)}$$

La dernière ligne de chaque variable *k* est calculée en fonction des sommes pondérées (par les coefficients du dénominateurs) du vecteur d'état à l'instant précédent et des *(N-1)*^{ères} lignes de la variable précédente. Elle dépend également des entrées courante et précédente. Au final, le nombre total d'opérations pour le schéma de Runge-Kutta 4 est le suivant :

Schéma	Additions	Multiplications	Décalages	Exp.	Mémoires
RK4	12*N*+1	9*N*+1	3*N*+1	/	*N*+1

Tableau 7 : Complexité combinatoire du schéma de Runge-Kutta d'ordre 4

VI.6.1.3 Comparaison

En regroupant les résultats des 3 tableaux, nous pouvons comparer la complexité combinatoire de chaque schéma en fonction de l'ordre du filtre considéré par rapport à celle obtenu avec un échantillonnage régulier. Le Tableau 8 montre l'évolution de la complexité combinatoire par rapport au cas régulier en fonction de *N*, lorsque l'ordre *N* devient grand.

Schéma	Additions	Multiplications	Décalages	Exp.	Mémoires
Bilinéaire	2,5	4,5	+*N*	/	1.5
RK4	6	4,5	+3*N*	/	1
RK23	7,5	11	+*N*/2	/	1.5
Résolution ED	3,25	5	/	+*N*	1.5

Tableau 8 : Augmentation de la complexité combinatoire par rapport au cas régulier en fonction de l'ordre *N* (pour *N* grand)

Nous remarquons que le nombre d'éléments mémorisant augmente de 50% par rapport au cas régulier lorsqu'une décomposition en filtre d'ordre 2 est nécessaire. En effet, comme chaque sortie dépend de l'entrée à l'instant précédent, *N/2* mémoires doivent être insérées pour sauvegarder les *N/2* sorties intermédiaires. Pour le schéma de Runge-Kutta RK4, le nombre de mémoire reste constant. Toutefois, il peut être divisé par 2, et ce quelle que soit la méthode, si la structure classique (forme directe 1) est utilisée. En revanche, comme pour les filtres FIR, le nombre d'additions et de multiplications est augmenté du fait de l'introduction de l'intervalle de

temps comme nouvelle variable dans l'équation de récurrence. Il est en effet normal d'avoir un surplus de calcul dans la mesure où il y a une donnée supplémentaire à traiter.

En comparant les méthodes entre elles, nous pouvons noter qu'en terme d'addition et de multiplication, les méthodes bilinéaires, RK4 et résolution de l'équation différentielle sont à peu près équivalentes. Toutefois, la résolution de l'équation différentielle qui requiert le calcul d'une exponentielle de matrice doit toujours calculer *N* exponentielles simples. Bien qu'il existe des techniques efficaces et rapides (du type Cordic) pour résoudre ce type de calcul, le coût est considérablement alourdi par rapport aux autres schémas. Enfin la méthode bilinéaire nécessite parmi les multiplications nécessaires, une division, dont la complexité est supérieure à une multiplication, ce qui dégrade également le coût par rapport au schéma de Runge-Kutta d'ordre 4.

VI.6.2 Architectures

D'après l'étude préliminaire des complexités combinatoires, seuls les schémas bilinéaires et RK4 sont intégrables en pratique. Nous proposons donc deux architectures spécifiques pour implémenter un filtre numérique utilisant ces méthodes. La description est réalisée au niveau conception.

VI.6.2.1 Architecture d'un filtre utilisant la méthode bilinéaire

Nous avons vu auparavant qu'un filtre utilisant la méthode bilinéaire devait être décomposé en filtres d'ordre 2 plus éventuellement un filtre d'ordre 1. Il existe donc deux structures élémentaires mises en cascade pour réaliser l'ensemble du filtre comme le montre la figure suivante.

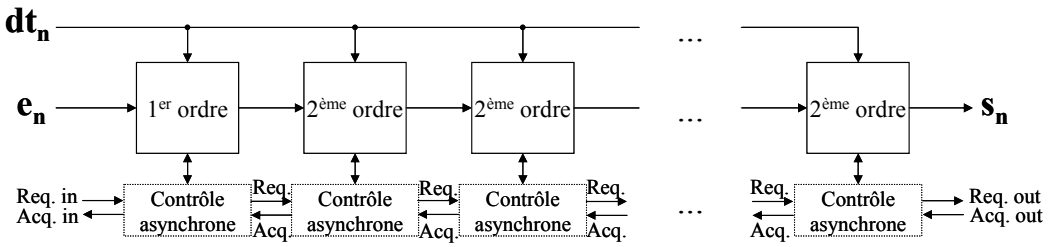


Figure 99 : Structure cascadée d'un filtre RII utilisant la méthode bilinéaire

Un contrôle asynchrone local pilote chaque bloc indépendamment. Lorsque le calcul d'une sortie de l'un des blocs est terminé, le contrôle envoie une requête au bloc suivant. Celui-ci peut alors recevoir les nouvelles données et acquitter.

L'équation (176) représentant le système d'état numérique peut se simplifier en calculant séparément les éléments des matrices Φ_n et Γ_n :

$$\begin{cases} x_{1,n} = \Phi_{11} x_{1,n-1} + \Phi_{12} x_{2,n-1} + \Gamma_1 \left(e_n + e_{n-1} \right) \\ x_{2,n} = \Phi_{21} x_{1,n-1} + \Phi_{22} x_{2,n-1} + \Gamma_2 \left(e_n + e_{n-1} \right) \\ s_n = c_1 x_{1,n} + c_2 x_{2,n} + D e_n \end{cases} \qquad \text{Eq. (190)}$$

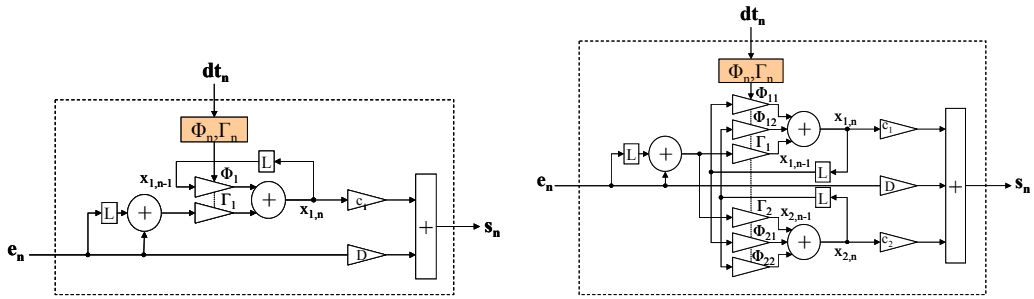


Figure 100 : Structures du 1er ordre et du 2ème ordre utilisant la méthode bilinéaire

Nous pouvons alors en déduire les deux structures élémentaires pour le 1er et 2ème ordre. Les architectures sont présentées sur la Figure 100 (les éléments mémorisants sont notés *L* pour ne pas être confondus à la matrice *D*).

VI.6.2.2 Architecture d'un filtre utilisant le schéma RK4

L'équation (187) montre que pour calculer chaque élément du vecteur d'état x_n, et par conséquent la sortie s_n, il faut au préalable, calculer chacune des variables k_1, k_2, k_3, k_4 correspondantes. Or, d'après les équations simplifiées (188) et (189), le $m^{\text{ième}}$ élément d'une variable k_i dépend du $(m+1)^{\text{ième}}$ élément de la variable k_{i-1} (k_1 dépendant de x_{n-1}), excepté pour le $(N-1)^{\text{ième}}$ éléments qui dépend des N éléments de la variable k_{i-1}. Nous proposons donc de concevoir une architecture composée de structures cascadées telle qu'elle est représentée sur la Figure 101. Comme chaque élément dépend du suivant, la cascade est inversée – l'étage *N-1* est donc placé en premier. Chaque structure calcule une sortie partielle et pointe vers une somme générale qui

calcule la valeur finale de la sortie. Il y a deux types de structure notées *A* et *B* correspondant respectivement aux équations (188) et (189) dont les architectures sont représentées sur la Figure 102 (le symbole α est utilisé pour remplacer le terme $dt_n/2$).

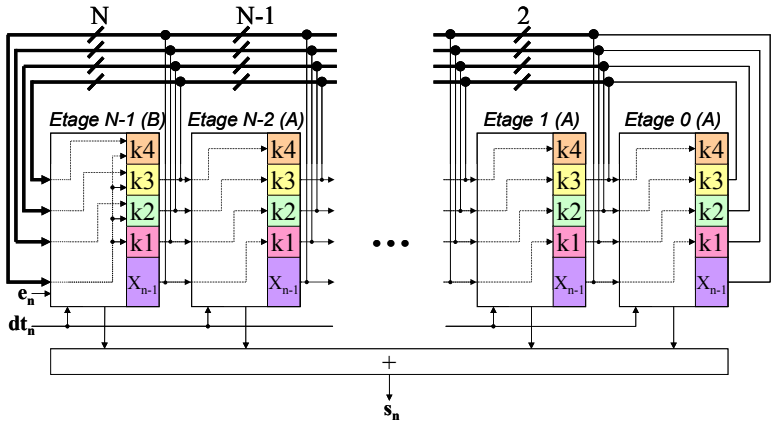


Figure 101 : Structure cascadée d'un filtre RII utilisant le schéma de Runge-Kutta d'ordre 4

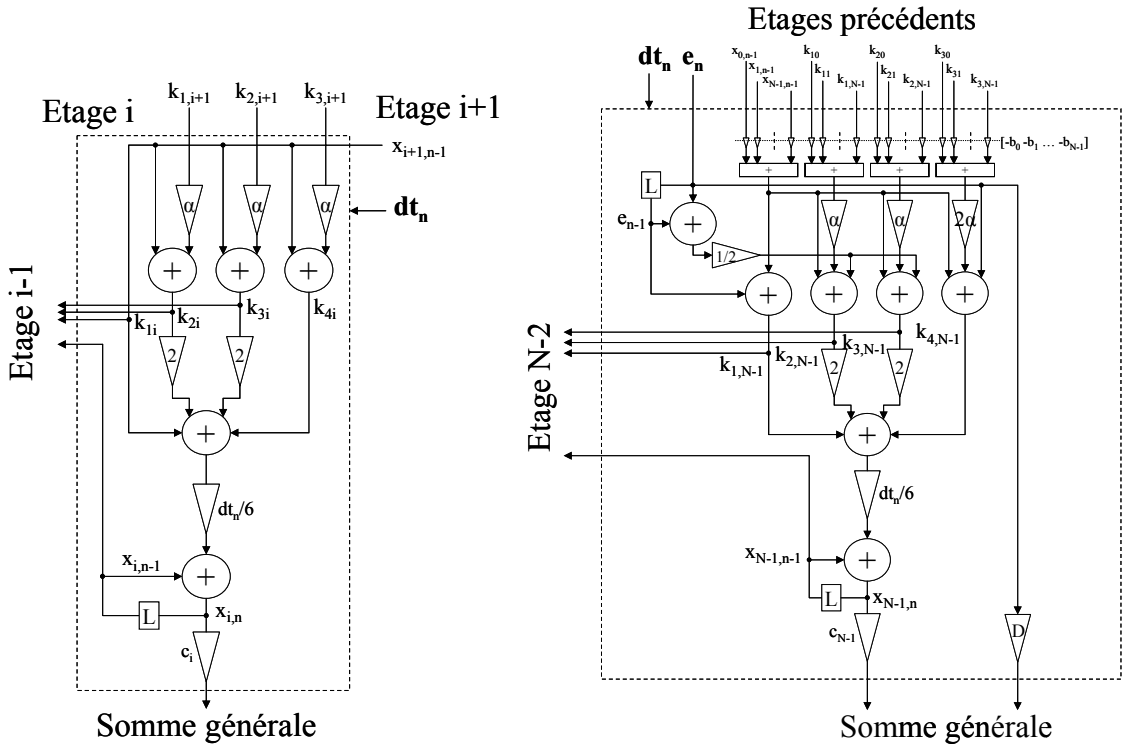


Figure 102 : Structures *A* et *B* utilisées par le schéma de Runge-Kutta d'ordre 4

Chaque structure possède un contrôle asynchrone local. Les connexions entre contrôles sont présentés respectivement pour les structures *A* et *B* sur les Figure 103 et Figure 104.

Les structures A reçoivent, de l'étage qui les précède, les variables x_{n-1}, k_1, k_2 et k_3 nécessaires au calcul de leurs propres variables k_1, k_2, k_3, k_4 et x_n. Pour un étage *i*, le contrôleur reçoit, dans un premier temps, une requête de l'étage *i+1* (*Req.* x_{n-1} *i+1*) afin que celui-ci lui transmette x_{n-1}. La structure calcule alors la valeur de k_1 et la transmet aux étages *i-1* et *N-1* après leur avoir envoyé une requête (*Req.* k_1 *i-1* et *Req.* k_1 *N-1*). Puis, le contrôleur reçoit, dans un second temps, une nouvelle requête de l'étage *i+1* (*Req.* k_1 *i+1*) afin qu'il lui transmette, cette fois-ci, k_1. La structure calcule alors la valeur de k_2 et la transmet aux étages *i-1* et *N-1* (*Req.* k_2 *i-1* et *Req.* k_2 *N-1*). Le processus est répété pour le calcul de k_3 et de k_4. Lorsque toutes les variables ont été calculées, la variable d'état est mise à jour puis stockée. La sortie de la structure peut alors être traitée; le contrôleur envoie une requête à l'additionneur (*Req. SOM*). Lorsqu'un nouvel échantillon arrive en entrée du filtre (*Req. in*), la variable d'état est envoyée à l'étage *i-1* (*Req.* x_{n-1} *i-1*) et à l'étage *N-1* (*Req.* x_{n-1} *N-1*).

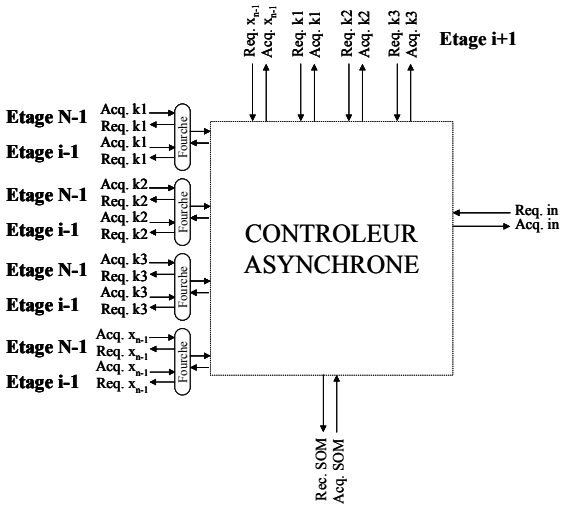


Figure 103 : Contrôleur asynchrone de la structure *A* utilisé par le schéma de Runge-Kutta d'ordre 4

Le fonctionnement de la structure *B* est similaire à celui d'une structure *A*. Lorsqu'un nouvel échantillon arrive, le contrôleur attend de recevoir toutes les variables d'état précédentes x_{n-1} pour calculer k_1 qu'il envoie à l'étage *N-2* et à lui-même (*Req.* k_1 *N-2* et *Req.* k_1 *N-1*). Puis, il recommence avec toutes les variables k_1 pour calculer k_2, etc.

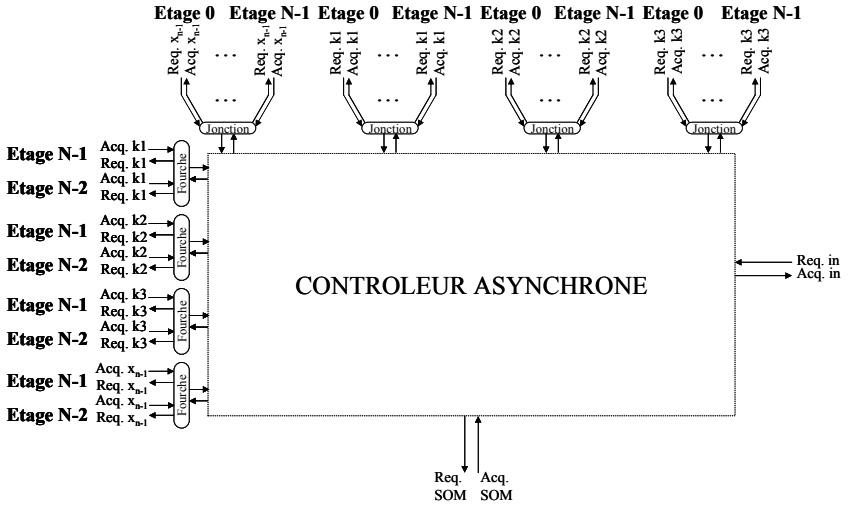


Figure 104 : Contrôleur asynchrone de la structure *B* utilisé par le schéma de Runge-Kutta d'ordre 4

Chaque structure pointe également vers la suivante pour transmettre ses variables x_{n-1}, k_1, k_2 et k_3 nécessaires au calcul de k_1, k_2, k_3 et k_4 de l'étage suivant. Or sur la structure B, il faut attendre que tous les éléments d'une variable aient été traités pour calculer la variable suivante. Ainsi pour que chaque étage ait ses 4 variables valides pour calculer sa sortie partielle, il faut passer 4 fois les *N* étages, donc 4 cycles sont nécessaires, comme le résume le Tableau 9. En rajoutant 2 cycles pour calculer la sortie, il faut au total 6 cycles pour traiter chaque échantillon en entrée.

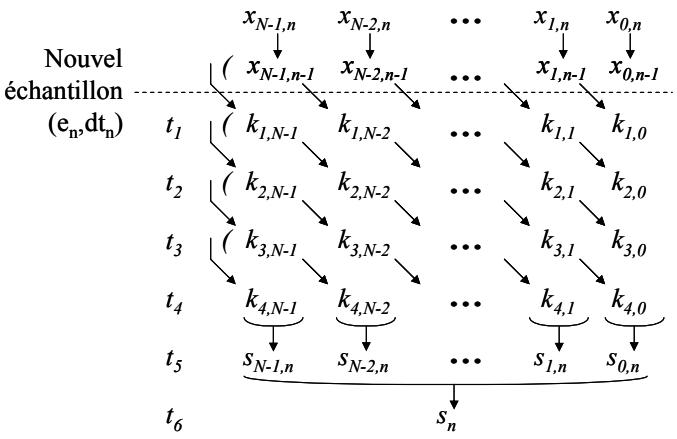


Tableau 9 : Cycles de calcul de l'architecture intégrant le schéma RK4

VI.7 Conclusion

Nous avons présenté le filtrage numérique à réponse impulsionnelle infinie d'un signal échantillonné non uniformément. L'équation de récurrence entre les échantillons d'entrée et les échantillons de sortie qui ne peut plus être définie par la transformée en Z, est obtenue par discrétisation d'un filtre analogique représenté dans l'espace d'état. Après avoir fait un état de l'art des schémas numériques standard (Euler, Bilinéaire et résolution directe de l'équation différentielle), nous avons généralisé la discrétisation de la représentation d'état en y intégrant des schémas évolués de type Runge-Kutta d'ordre 3 et d'ordre 4. Nous avons étudié leur stabilité vis-à-vis des paramètres du filtre et nous en avons déduit un flot de conception prenant en compte les spécificités de chaque schéma. Nous avons formulé, en outre, des hypothèses sur la réponse en fréquence du filtre numérique, qui dans le cadre d'un échantillonnage par traversée de niveaux n'est pas déformée du fait du suréchantillonnage (c'est-à-dire des fréquences d'échantillonnage élevées par rapport à la bande utile du filtre qui repousse les déformations vers les hautes fréquences).

Une série non exhaustive de simulations est ensuite présentée. Les exemples de filtrage illustrent les études effectuées précédemment sur la stabilité et les déformations de la réponse en fréquence. Ils nous permettent de restreindre l'étude des architectures matérielles aux seuls schémas suffisamment efficaces en laissant de coté les méthodes d'Euler.

Puis, nous avons étudié l'implémentation matérielle de chaque schéma. L'analyse préliminaire de leurs complexités combinatoires nous a permis de réduire le nombre de schémas à deux : la méthode bilinéaire, inconditionnellement stable et la méthode de Runge-Kutta d'ordre 4, conditionnellement stable. Approximativement, ces deux méthodes nécessitent cinq fois plus de multiplications et d'additions que dans le cas régulier. Nous avons alors proposé deux architectures asynchrones que nous décrivons : la première est issue d'une décomposition du filtre global en filtres d'ordre 1 et d'ordre 2 cascadés. La seconde est directement un filtre d'ordre N composé de structures bouclées. Dans les deux cas, il est important de noter que le nombre de cycles nécessaires au calcul d'un échantillon en sortie est constant ce qui rend l'utilisation d'une représentation d'état numérique et de ces architectures dédiées très intéressantes pour le filtrage RII – voire même le filtrage en général – de signaux échantillonnés non uniformément.

CHAPITRE VII

Evaluation d'une chaîne de traitement numérique du signal asynchrone

---

Ce chapitre est dédié à l'évaluation d'une chaîne de traitement numérique du signal asynchrone. Dans un premier temps, un exemple est considéré pour illustrer la mise en œuvre d'une chaîne. Il concerne le filtrage d'un signal de parole dont l'activité est réduite à 25% du temps total (75% de silence).

Dans un second, nous définissons un critère général de comparaison entre une chaîne synchrone et asynchrone. Il permet de déterminer la technologie à privilégier en fonction du traitement considéré. Dans chaque cas, la charge de calcul totale, utilisée pour traiter une série de points échantillonnés pendant une durée finie, est prise en compte. Comme un traitement asynchrone est plus complexe que le même traitement réalisé dans le cas synchrone, il doit, pour être avantageux, s'appliquer sur un nombre de points réduit afin que, sur une durée donnée, la charge de calcul totale soit plus faible que dans le cas synchrone – le nombre limité de points compensant une complexité accrue du traitement de chaque point.

Des résultats numériques sont ensuite présentés dans le cadre des filtres mis en œuvre dans les chapitres précédents. Ils montrent qu'à partir d'un rapport minimal entre le nombre de points de la chaîne synchrone, échantillonnés régulièrement, et le nombre de points de la chaîne asynchrone, échantillonnés non uniformément, un traitement asynchrone est préconisé quel que soit l'ordre du filtre considéré; la charge de calcul totale sur une durée limitée qui dans tous les cas est plus faible, induit une réduction de l'activité de la chaîne et donc de sa consommation électrique.

VII.1 Choix d'une application

Le traitement numérique d'un signal échantillonné non uniformément par traversée de niveaux est adapté au signal. Les performances de la chaîne de traitement dépendent donc du choix de l'application. Dans ce travail de thèse, nous n'avons pas particulièrement ciblé un signal ou une application donnée. En effet, en étudiant le filtrage numérique de signaux échantillonnés non uniformément, nous sommes restés volontairement dans le cadre général afin d'analyser les algorithmes dans leur ensemble. Cependant pour évaluer une chaîne de traitement, nous allons définir une application. Celle-ci doit, d'une part, nous permettre d'illustrer les traitements définis dans les chapitres précédents sur un signal réel et de comparer les performances avec une chaîne synchrone et d'autre part, nous conduire à la définition d'un critère quant au choix de la technologie à utiliser en fonction de l'application.

Nous décidons donc de traiter un signal de parole. Celui-ci est extrait d'une conversation entre deux interlocuteurs. En effet, il a été montré statistiquement que le temps de parole lors d'un dialogue n'était que de 25% pour chaque interlocuteur – 50% du temps total de la discussion étant composé de silence [Fontolliet 1983]. Ce type de signal est caractéristique de ce que nous voulons mettre en œuvre puisqu'une chaîne de traitement synchrone utilisant un échantillonnage uniforme fonctionnera tout le temps alors qu'une chaîne de traitement asynchrone ne fonctionnera pas pendant 75% du temps (en ne considérant que l'un des deux interlocuteurs).

L'application que nous avons voulu mettre en œuvre, consiste en un filtrage passe-bas du signal afin de détecter la fréquence fondamentale de la voix de l'un des interlocuteurs. Bien qu'il existe d'autres traitements dédiés à cette tâche, nous rappelons qu'il s'agit simplement dans notre exemple de réaliser une opération de filtrage.

Une fois l'application et le signal connu, nous définissons le gabarit du filtre à implémenter. Il est connu que la fréquence fondamentale d'un homme se situe dans l'intervalle *[150Hz; 200Hz]* tandis celle d'une femme appartient à l'intervalle *[200Hz; 300Hz]* [Rabiner *et al.* 1978]. Le signal de parole est supposé à bande limitée de fréquence maximale $F_{max} = 4kHz$. Un signal échantillonné régulièrement à la fréquence $F_e = 8kHz$ sera donc utilisé comme référence afin de comparer les approches synchrone et asynchrone.

VII.2 Mise en œuvre d'une chaîne de traitement asynchrone

VII.2.1 Conversion Analogique/Numérique Asynchrone

La conversion analogique-numérique d'un signal de parole a été étudiée par Emmanuel Allier lors de ses travaux de thèse [Allier 2003]. Il a spécifié les caractéristiques internes du convertisseur qu'il faut utiliser pour échantillonner non uniformément un signal de parole avec une résolution effective de 8 bits. Les caractéristiques sont rappelées dans le tableau suivant :

Résolution effective : *ENOB*	8 bits
Dynamique du signal : Δv_{in}	$V_{alim}/2$ centré en $V_{alim}/2$
Fréquence du timer : F_C	1,5MHz
Temps de boucle : δ_{MAX}	100ns
Profondeur du timer : M_{TIMER}	10 bits
Nombre de niveaux : *N*	15

Tableau 10 : Cahier des charges du CANA pour une application de parole

Ces paramètres seront utilisés dans nos modèles pour échantillonner le signal de parole par traversée de niveaux.

VII.2.2 Conception du filtre numérique

La conception d'un filtre numérique doit prendre en compte les caractéristiques spécifiées par un cahier des charges. Celles-ci sont représentées dans un gabarit qui situe les fréquences de coupures et les gains associés. Nous posons alors les paramètres suivants :

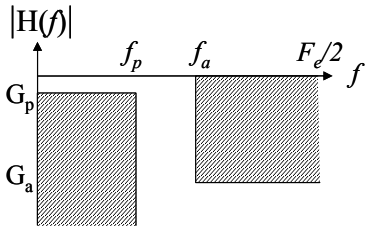


Figure 105 : Gabarit du filtre passe bas

- Fréquence de coupure : $f_p = 300Hz$,
- Fréquence minimale de la bande atténuée : $f_a = 400Hz$,
- Ondulation dans la bande passante : $G_p = 1dB$,
- Atténuation minimale de la bande atténuée : $-G_a = 40dB$.

VII.2.2.1 Implémentation d'un filtre RIF

Nous implémentons dans un premier temps un filtre numérique à réponse impulsionnelle finie utilisant un produit de convolution asynchrone d'ordre 0. Le filtre, répondant aux caractéristiques présentées précédemment, est de type optimal equiripple à 115 coefficients. La réponse en fréquence est représentée sur la Figure 106.

VII.2.2.2 Implémentation d'un filtre RII

Dans un second temps, nous implémentons un filtre numérique à réponse impulsionnelle infinie basé sur la discrétisation de la représentation d'état du filtre analogique équivalent. Conformément aux conclusions émises dans le chapitre précédent, seuls les schémas numériques de type bilinéaire (inconditionnellement stable) et de type Runge-Kutta d'ordre 4 – RK4 – (conditionnellement stable) sont utilisés. Toutefois, dans les deux cas, les matrices A, B, C et D de la représentation d'état sont communes aux deux méthodes. Celles-ci sont obtenues à l'aide de fonctions d'approximation. Nous choisissons d'utiliser un polynôme de Butterworth d'ordre 19. Les pôles et la réponse en fréquence sont représentés sur la Figure 107. Le sous-espace du plan complexe analogique, définissant la condition limite de stabilité pour la méthode RK4, est fixé par l'intervalle de temps le plus grand qui est imposé par la profondeur du timer. Nous pouvons remarquer que tous les pôles sont inclus dans le sous-espace; le filtre numérique sera donc stable.

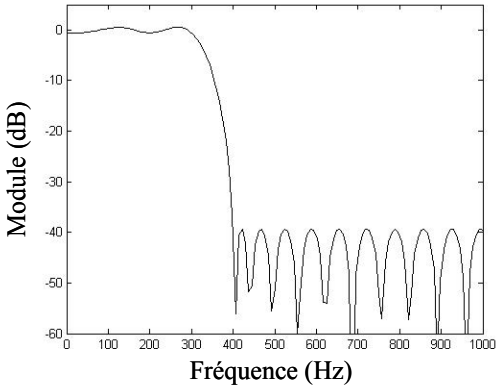


Figure 106 : Réponse en fréquence du filtre RIF

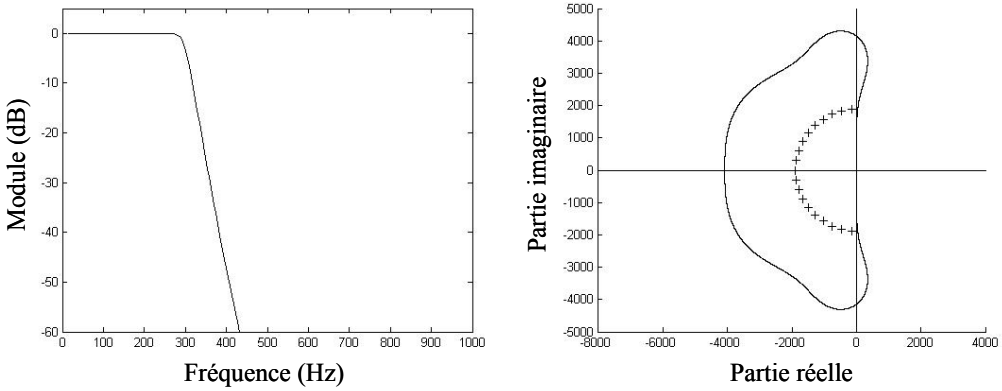


Figure 107 : Réponse en fréquence (à gauche) et pôles (à droite) du filtre RII

VII.2.3 Résultats

Nous avons donc simulé le filtrage d'un signal de parole échantillonné non uniformément sur 15 niveaux. Les résultats, concernant une période active d'une durée de 1,3 seconde, sont présentés sur la Figure 108. Nous pouvons observer que globalement, les résultats sont très similaires. Cette remarque se vérifie en calculant la transformée de Fourier du signal reconstruit à l'ordre 0 sur chaque sortie centrée sur l'une des voyelles filtrées.

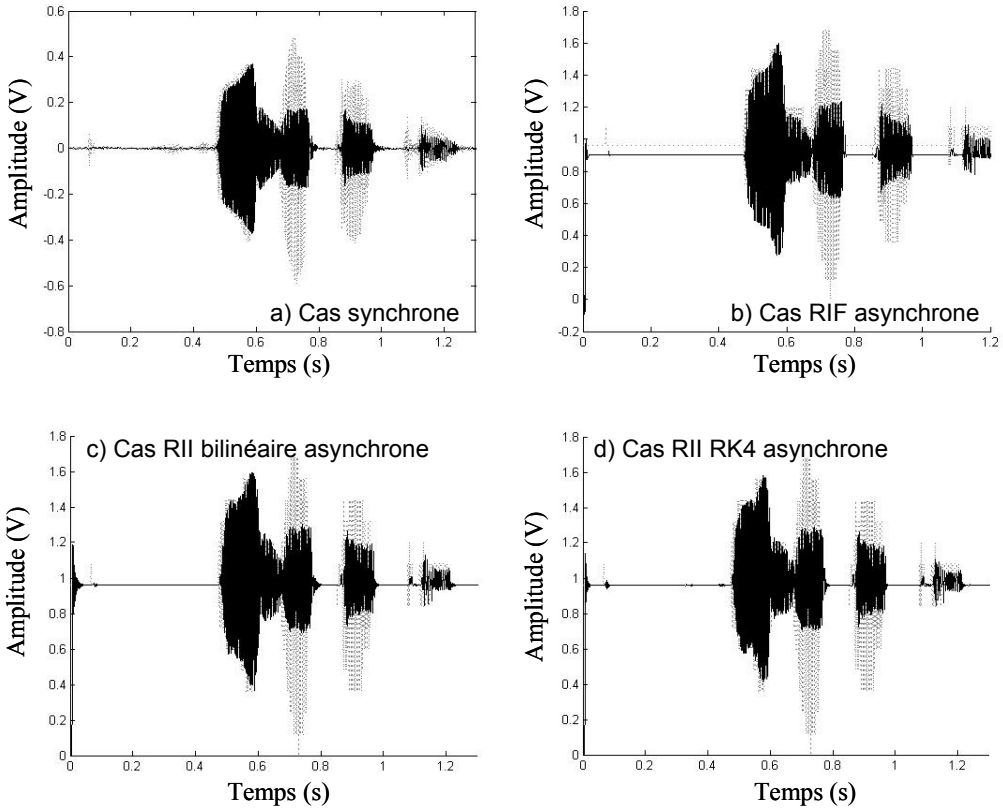


Figure 108 : Filtrage numérique d'un signal de parole échantillonné non uniformément par traversée de niveaux

La Figure 109 montre en effet que les quatre sorties ont toutes une seule composante sinusoïdale à la fréquence *fo = 265Hz* environ, en négligeant les ondulations introduites par le fenêtrage temporel et le pas fréquentiel. Nous pouvons alors en déduire que l'interlocuteur dont la voix a été acquise, échantillonnée puis filtrée, est une femme.

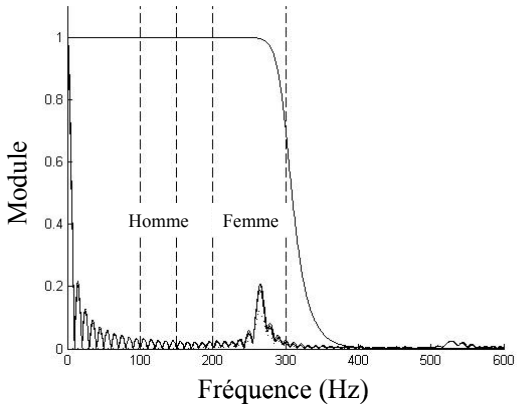


Figure 109 : Transformées de Fourier des signaux reconstruits à l'ordre 0

Dans un deuxième temps, nous nous intéressons à la complexité combinatoire des différents processus de filtrage afin de déterminer comment l'activité de la chaîne de traitement s'adapte à l'activité du signal. Dans le cadre des filtres à réponse impulsionnelle infinie, il n'est pas possible d'analyser l'évolution temporelle de la charge de calcul puisque par définition, ils ont été conçus pour traiter chaque nouvel échantillon entrant avec un nombre d'opérations fixe. En revanche, les filtres à réponse impulsionnelle finie utilisent un produit de convolution asynchrone qui est réalisé grâce à un algorithme itératif. Nous avons vu dans le chapitre VI que le nombre d'itérations dépendait, pour un filtre donné, c'est-à-dire pour un nombre de coefficients donné, du rapport entre la période d'échantillonnage de la réponse impulsionnelle et la valeur moyenne des intervalles de temps du signal d'entrée, localisée pendant la durée totale de la réponse impulsionnelle. Or, en échantillonnant un signal par traversée de niveaux, la valeur des intervalles de temps est une image directe de l'activité du signal. Ainsi, en étudiant l'évolution temporelle du nombre d'itérations de l'algorithme (partie gauche de la Figure 110), voire le nombre d'échantillons d'entrée utilisé pour le calcul d'une sortie (partie droite de la Figure 110), nous pouvons analyser les relations entre l'activité du traitement et l'activité du signal. A première vue, il faut noter que les variations des deux courbes sont identiques. Ceci est tout à fait normal puisque le nombre d'itérations, nécessaire au calcul d'une sortie, est approximativement égal à la somme du nombre de coefficients de la réponse impulsionnelle (fixe) et du nombre d'échantillons d'entrée utilisés (variable). Puis, nous pouvons constater que la charge de calcul du filtre est parfaitement adaptée à l'activité du signal. Les zones dans lesquels la charge est maximale correspondent notamment au traitement des voyelles dont les intervalles de temps sont les plus faibles. Le reste du temps, il faut noter que la charge est constante. Ceci s'explique par la saturation du timer. En effet, lorsque les variations du signal deviennent plus petites que la valeur du quantum du convertisseur, les intervalles de temps sont trop grand vis-à-vis du timer qui sature et qui doit alors échantillonner le signal d'entrée. Durant toute cette plage d'inactivité, le signal est échantillonné uniformément. Dans le morceau de signal considéré, cette situation a principalement lieu au cours de l'échantillonnage des terminaisons de consonnes.

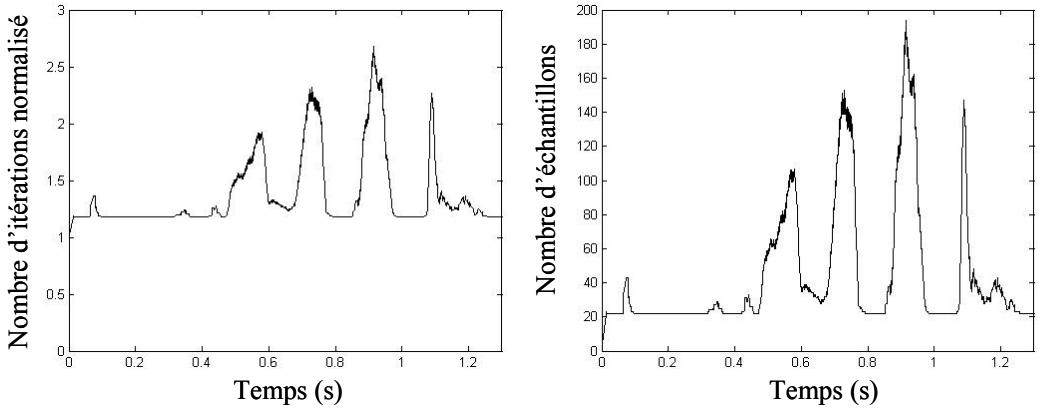


Figure 110 : Evolutions temporelles du nombre d'itérations et d'échantillons d'entrée utilisés dans le produit de convolution asynchrone.

VII.3 Critère de choix de la technologie

L'exemple que nous venons de présenter nous a permis d'introduire le traitement d'un signal à activité variable par une chaîne asynchrone. Jusqu'à présent, le traitement numérique du signal est essentiellement basé sur un échantillonnage régulier et une conception synchrone des éléments de la chaîne. Nous avons vu précédemment que la prise en compte du signal était primordiale pour diminuer l'activité de la chaîne, donc sa consommation, ce que ne fait pas le cas synchrone. Or, nous avons vu également que le traitement d'un signal échantillonné non uniformément était plus complexe qu'un signal échantillonné régulièrement du fait de l'introduction de l'intervalle de temps dans le calcul. La solution asynchrone n'est donc pas forcément toujours la meilleure.

C'est pourquoi, nous proposons de définir un critère sur le choix de la technologie à préconiser. Les deux principales différences entre une chaîne de traitement synchrone et asynchrone étant la complexité du calcul et le principe d'échantillonnage, c'est-à-dire le nombre de points échantillonnés, le critère naturel est le suivant :

$$G_{T,\varphi} = \frac{f_{sync}(\varphi) \cdot N_{sync}}{f_{async}(\varphi) \cdot N_{async}} \qquad \text{Eq. (191)}$$

Pour un traitement φ donné, f_{sync} et f_{async} représentent respectivement un modèle de la complexité de calcul de φ dans le cas synchrone et asynchrone. Ce modèle peut avoir plusieurs niveaux

d'abstraction. Ainsi, dans notre cas, nous nous limitons simplement aux nombres total d'additionneurs et multiplieurs nécessaires. Dans le futur, le modèle pourra être affiné lors de la conception microélectronique en y intégrant tous les opérateurs, en les pondérant en fonction de leurs complexités voire en utilisant l'implémentation matérielle et ses caractéristiques (rapidité, consommation, surface, …). Pour une fenêtre temporelle T, N_{sync} et N_{async} représentent le nombre de points échantillonnés régulièrement et non uniformément sur cette durée. $G_{T,\varphi}$ est donc le rapport entre le nombre total de calculs sur la durée T dans le cas synchrone sur le nombre total de calculs sur la même durée T dans le cas asynchrone. Si $G_{T,\varphi} < 1$, une chaîne synchrone sera plus performante pour réaliser le traitement φ, tandis que si $G_{T,\varphi} > 1$, la chaîne asynchrone sera plus performante.

Pour les traitements proposés dans cette thèse, les critères sont donc les suivants :

$$G_{T,RIF} = \frac{2N \cdot N_{sync}}{4N\left(1+\frac{N_{async}}{N_{sync}}\right)\cdot N_{async}} = \frac{1}{2\left(1+\frac{N_{async}}{N_{sync}}\right)}\frac{N_{sync}}{N_{async}} \qquad \text{Eq. (192)}$$

Pour un filtre à réponse impulsionnelle finie, le rapport entre les complexités est indépendant du nombre de coefficient de la réponse impulsionnelle. En revanche, pour un filtre à réponse impulsionnelle infinie, les rapports dépendent de l'ordre N du filtre :

$$G_{T,RII\ bilinéaire} = \frac{\left(2N+1+2N\right)\cdot N_{sync}}{\left(5N+9N\right)\cdot N_{async}} = \frac{4N+3}{14N}\frac{N_{sync}}{N_{async}} \qquad \text{Eq. (193)}$$

$$G_{T,RII\ RK4} = \frac{\left(2N+1+2N+2\right)\cdot N_{sync}}{\left(12N+1+9N+1\right)\cdot N_{async}} = \frac{4N+3}{21N+1}\frac{N_{sync}}{N_{async}} \qquad \text{Eq. (194)}$$

Nous avons donc calculé l'évolution du gain de chaque traitement en fonction du rapport entre le nombre d'échantillons synchrones et asynchrones. Les résultats sont présentés sur la Figure 111. Pour les filtres à réponse impulsionnelle infinie de type bilinéaire (trait plein) et de type Runge-Kutta d'ordre 4 (pointillé), les courbes sont paramétrées par l'ordre du filtre N. Plus l'ordre augmente plus les courbes tendent vers une droite limite. En revanche pour le filtre à réponse impulsionnelle finie, il n'y qu'une seule courbe (alternance trait point). Pour tous les traitements, la logique est respectée : comme ils sont plus compliqués dans le cas asynchrone que dans le cas

synchrone, il faut que le nombre de points traités dans le cas asynchrone soit beaucoup plus faible pour que l'utilisation de cette technologique ait un intérêt.

Pour les filtres à réponse impulsionnelle infinie, le rapport minimum entre le nombre de points asynchrones traités et le nombre de points synchrones est variable tandis que pour le filtre à réponse impulsionnelle finie, le rapport est fixe. Nous avons donc réalisé pour chaque traitement une coupe des courbes de gains en $G_{T,\varphi} = 1$, en fonction de l'ordre pour déterminer le rapport minimum à partir duquel l'approche asynchrone est favorable. Ces coupes sont représentées sur la Figure 112 (la légende de la Figure 111 est conservée). Les rapports augmentent avec l'ordre et tendent vers des valeurs limites à partir desquelles la technologie asynchrone est systématiquement privilégiée quel que soit l'ordre du filtre. Ces valeurs sont résumées sur le Tableau 11.

Traitement	RIF	RII Bilinéaire	RII RK4
Rapport	2,7	3,5	5,25

Tableau 11 : Rapport minimum entre le nombre de points synchrones et le nombre de points asynchrones privilégiant la technologie asynchrone quel que soit l'ordre du filtre

Le choix de la technologie dépend donc du rapport entre le nombre de points échantillonnés dans chaque cas, donc du processus d'échantillonnage. Cependant, il n'est pas possible de relier directement ce rapport au signal et plus particulièrement à son activité. En effet, dans le cas synchrone, le nombre d'échantillons est soit fixé par la fréquence maximale du signal, soit par des contraintes matérielles qui imposent la fréquence d'échantillonnage. Dans le cas asynchrone, le nombre d'échantillons dépend à la fois des variations du signal et du convertisseur (nombre de niveaux, profondeur du timer). Or, l'échantillonnage par traversée de niveaux ne génère pas forcément le même nombre de points pour deux signaux actifs différents; seules les zones inactives sont échantillonnées de la même manière. Le critère décrit par l'équation (191) est par conséquent le critère le plus précis que l'on puisse définir, bien que le signal n'apparaisse pas clairement dans la formulation.

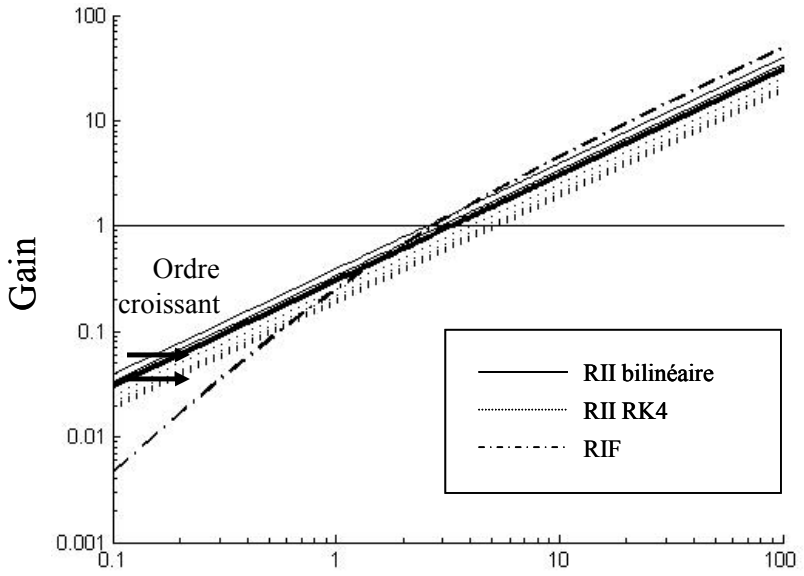


Figure 111 : Gain d'une chaîne de traitement numérique asynchrone en fonction du rapport entre le nombre d'échantillons synchrones et asynchrones. Pour le filtrage RII, les courbes de gains sont paramétrées par l'ordre du filtre. Il y a donc une série de courbes pour chaque méthode; les courbes tendent toutes vers une droite limite.

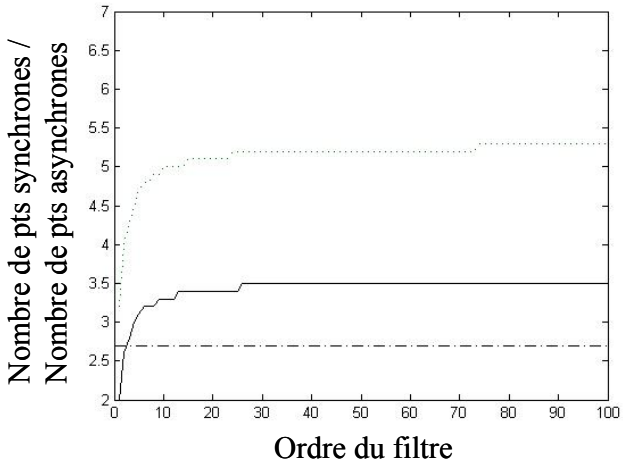


Figure 112 : Rapport minimum entre le nombre de points synchrones et le nombre de points asynchrones en fonction de l'ordre des filtres

Cependant, nous pouvons autoriser une exception lorsque le signal est déjà échantillonné. En effet, si l'on considère l'inactivité du signal comme du silence, il est possible de faire varier cette durée tandis que la zone active reste identique, c'est-à-dire échantillonnée de la même manière. Ainsi, dans le cas synchrone, le silence est toujours échantillonné régulièrement selon la fréquence d'échantillonnage tandis que dans le cas asynchrone, le silence est échantillonné régulièrement à cause de la saturation du timer. Dans l'exemple présenté au début du chapitre, nous avions supposé que le signal de parole était inactif pendant 75% de la durée totale du signal. En conservant la durée active de parole constante, nous avons fait varier la plage de silence entre 0% et 99% du temps total afin d'obtenir une extrapolation du critère sur l'activité du signal. Nous tenons bien à préciser que les résultats présentés sur la Figure 113 et le Tableau 12 ne sont valables que dans le cas particulier du signal considéré. Ils permettent simplement d'étudier le gain de la technologie asynchrone sur la technologie synchrone si le même signal avait eu une activité différente.

Par ailleurs, la saturation du timer a pour conséquence que les intervalles de temps sont toujours inférieurs ou égal à une valeur limite. Compte tenu de cette valeur, tous les filtres considérés sont stables même celui utilisant le schéma RK4. Cependant, lorsque le signal est constant, une série de points identiques est prélevée; le traitement devient redondant ce qui induit une activité accrue de la chaîne. Or dans le cas asynchrone, comme les communications entre blocs fonctionnels sont basées sur des protocoles de requête-acquittement, il est possible de ne pas traiter les points échantillonné lors de la saturation du timer. En effet, lorsqu'un point est échantillonné, une requête est envoyée à l'étage de traitement. Ainsi, en détectant la saturation à l'aide du signal interne au convertisseur prévu à cet effet, le convertisseur peut ne pas envoyer de requête. Si cela ne pose aucun inconvénient pour le filtre RIF, le stockage de la variable d'état induit une erreur sur les filtres RII car elle ne correspond plus au calcul du point précédent mais à celui qui précédait la période de saturation. C'est pourquoi, ce principe n'a pas été évoqué plus tôt et qu'il est introduit ici à titre indicatif.

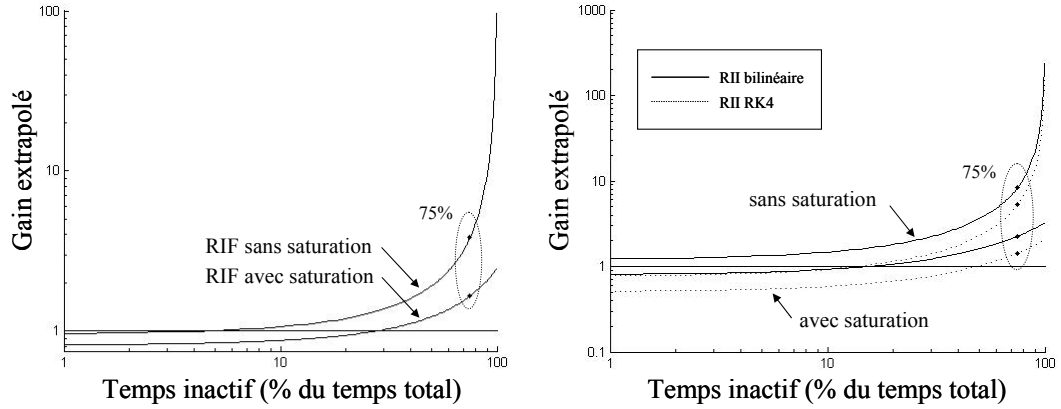


Figure 113 : Gains extrapolés en fonction de l'activité du signal de parole présenté dans notre exemple pour les différents filtres

Traitement		G_T min (1%)	G_T à 75%	G_T max (99%)	Seuil G_T = 1
RIF	Avec saturation	0,81	1,65	2,45	28%
	Sans saturation	0,96	3,83	95,8	5%
RII bilinéaire	Avec saturation	0,81	2,26	3,22	16%
	Sans saturation	1,22	8,41	239	/
RII RK4	Avec saturation	0,53	1,49	2,11	46%
	Sans saturation	0,81	5,54	157	12%

Tableau 12 : Gains extrapolés en fonction de l'activité du signal de parole présenté dans notre exemple pour les différents filtres

Pour le signal de parole considéré précédemment, nous pouvons noter que lorsqu'il présente une inactivité égale à 75% du temps total, l'utilisation d'une chaîne de traitement asynchrone du signal est préférable – son activité étant réduite approximativement d'un facteur 2. Il est également intéressant de remarquer que pour tous les traitements, excepté le filtre RII bilinéaire sans saturation du timer, il existe un seuil à partir duquel l'approche synchrone doit être privilégiée – seuil qui est réduit si le traitement des points issus de la saturation du timer n'est pas considéré.

Sans le prouver formellement, cette extrapolation permet de montrer que plus un signal possède de longue période d'inactivité, plus le rapport entre le nombre de points synchrones et asynchrones augmente et plus l'approche asynchrone doit être préconisée pour le traitement numérique du signal.

VII.4 Conclusion

Ce chapitre était dédié à l'évaluation d'une chaîne de traitement numérique asynchrone – chaîne composée d'un convertisseur analogique-numérique à échantillonnage par traversée de niveaux et d'un étage de filtrage – comparativement à une chaîne similaire synchrone basée sur un échantillonnage régulier. Nous avons défini un critère permettant, pour un traitement donné, de choisir entre les deux approches possibles en prenant en compte, dans chaque cas, la charge de calcul totale utilisée pour traiter une série de points échantillonnés pendant une durée finie. Les traitements asynchrones qui doivent utiliser l'information portée à la fois par l'amplitude et l'intervalle de temps des points, sont plus complexes que le même traitement réalisé dans le cas synchrone. Pour être avantageux, ils doivent s'appliquer sur un nombre de points réduits afin que, sur une durée donnée, la charge de calcul totale soit plus faible que dans le cas asynchrone.

Par ailleurs, une application a été proposée pour illustrer une chaîne de traitement. Les résultats montrent qu'à partir d'une durée d'inactivité, le nombre de points échantillonnés non uniformément était suffisamment réduit pour compenser l'augmentation de la complexité des calculs (28% du temps total pour le filtre RIF, 16% pour le filtre RII utilisant la méthode bilinéaire et 46% pour le filtre RII utilisant le schéma RK4) et rendre attractif l'approche asynchrone. Bien que ces seuils minimaux d'inactivité soient propres aux signaux considérés, ils montrent que plus un signal est inactif, plus l'adoption d'une chaîne de traitement numérique asynchrone est préconisée car la réduction significative de son activité contribuera à une réduction du même ordre de grandeur de la consommation électrique.

CONCLUSION

Ce travail de thèse se situe dans le contexte des Systèmes sur Puces (SoC) et des Objets Communicants. Il propose une réflexion visant à repenser complètement les chaînes de traitement numérique du signal afin de réduire leur activité et par conséquent la consommation électrique des systèmes dans lesquels elles sont intégrées. Les études ont donc été menées au niveau de la théorie du signal, de l'algorithmique des traitements, et de l'implémentation matérielle. Le principe fondamental de cette nouvelle catégorie de chaîne de traitement du signal est de concevoir des systèmes uniquement contrôlés par l'information du signal d'entrée, en combinant une conception matérielle asynchrone (sans horloge globale) et un échantillonnage non uniforme dans le temps adapté au signal. La nouvelle chaîne est appelée « *asynchrone* » par opposition aux chaînes « *synchrones* » basées sur une conception matérielle synchrone et sur un échantillonnage régulier.

Le premier élément de la chaîne mis en jeu est le Convertisseur Analogique/Numérique Asynchrone (CANA). L'étude complète de cet étage a été effectuée au cours d'un travail de thèse précédent [Allier 2003]. Ce convertisseur réalise un échantillonnage non uniforme dit « *par traversée de niveaux* ». Pour convertir un signal analogique, des niveaux ont été disposés régulièrement le long de la dynamique du signal. Un échantillon est alors prélevé lorsque le signal analogique croise l'un des niveaux de référence. Le signal est non uniformément échantillonné car le processus dépend de ses variations. L'information temporelle est donc mesurée; le système de conversion délivre alors en sortie des couples amplitude et intervalle de temps. L'échantillonnage régulier, défini par le théorème de Shannon, est en pratique réalisé par un système dont l'horloge est beaucoup plus grande que la limite théorique fixée par le double de la fréquence maximale du signal d'entrée. Ceci entraîne donc un nombre important d'échantillons redondants possédant peu d'information et induit une suractivité de la chaîne de traitement et une consommation électrique accrue. L'échantillonnage par traversée de niveaux se démarque donc du cas classique car seuls des échantillons pertinents sont prélevés. Le nombre de points stockés est réduit, tout comme l'activité du circuit.

Du point de vue matériel, l'architecture du convertisseur est conçue en logique asynchrone. L'horloge globale pilotant les différents blocs internes a été supprimée au profit de signaux de communication localisés. Les échanges d'information s'effectuent alors à l'aide de requêtes et

d'acquittements entre blocs fonctionnels grâce à des protocoles de type « poignée de main ». Proposant des systèmes contrôlés par des événements irréguliers, la technologie asynchrone est parfaitement adaptée à l'échantillonnage par traversée de niveaux. En prévision d'une implémentation matérielle prochaine des traitements présentés dans ce manuscrit, les architectures proposées prennent en compte la spécificité d'un contrôle localisé sans toutefois détailler explicitement leur mise en œuvre.

La chaîne de traitement numérique du signal asynchrone – Conversion Analogique/Numérique Asynchrone, traitement spécifique, Conversion Numérique/Analogique Asynchrone – étant définie, nous avons commencé notre travail en analysant d'une part la pertinence de l'information portée par les échantillons prélevés sur le signal d'entrée et en étudiant d'autre part la possibilité d'analyser le contenu spectral du signal analogique à l'aide des échantillons. Les deux questions sont liées au schéma d'échantillonnage c'est-à-dire à la transformée de Fourier de la fonction d'échantillonnage (i.e. du peigne de Dirac discrétisant le signal analogique). Le processus d'échantillonnage par traversée de niveaux étant asservi aux évolutions du signal d'entrée, nous avons tout d'abord analysé le schéma caractéristique des signaux périodiques. Nous avons alors montré que le schéma d'échantillonnage est un spectre composé de raies dont la fréquence fondamentale est la même que le signal analogique. Le spectre du signal échantillonné est alors replié. En étendant l'analyse du schéma d'échantillonnage aux signaux non périodiques dont les signaux impulsionnels, nous avons montré que la fonction d'échantillonnage est non stationnaire : ses propriétés fluctuantes sont dans le temps. Le schéma d'échantillonnage est alors étudié avec une analyse temps-fréquence et conduit, une nouvelle fois, à un repliement local du spectre. Aucune information sur le signal analogique ne peut être donc déduite à partir du spectre du signal échantillonné. Ceci vient du fait que la transformée de Fourier Discrète du signal échantillonné ne tient pas compte explicitement de la distance entre deux points car elle ne traite que l'amplitude des échantillons, en leur attribuant la même valeur informative. Or, avec un échantillonnage par traversée de niveaux, chaque intervalle de temps est contient une information importante sur le signal. En effet, si l'intervalle de temps est grand par exemple, cela veut dire qu'il y a peu de points dans son entourage et qu'il est donc singulier. Bien qu'elle ne nous intéresse pas en terme de donnée à traiter, l'information contenue dans cet intervalle est significative : *« L'amplitude du signal n'a pas évolué de plus d'un quantum depuis très longtemps ».* Lors de variations du signal analogique, les intervalles sont plus petits et de mêmes grandeurs entre eux. Dans cette configuration, l'information des intervalles de temps sert à

pondérer chaque amplitude. Par conséquent, cette étude préliminaire nous a permis de conclure que pour traiter un signal échantillonné par traversée de niveaux, il faut concevoir des algorithmes traitant explicitement la valeur des intervalles de temps en plus des amplitudes. Ainsi, en nous intéressant à la reconstruction pratique d'un signal analogique (rôle du Convertisseur Numérique/Analogique Asynchrone), nous avons montré que le spectre du signal reconstruit par un bloqueur d'ordre 0 a la même formulation que la transformée de Fourier du signal échantillonné (à un terme de gain et de phase près) dans laquelle chaque amplitude est pondérée par son intervalle de temps. La pondération des amplitudes a permis d'éviter le repliement. En revanche, elle introduit une distorsion qui peut toutefois être atténuée par un filtre de lissage.

Fort de ce constat, nous avons alors entrepris de traiter un signal échantillonné non uniformément en considérant le filtrage car il s'agit d'une opération élémentaire en traitement du signal, aussi bien pour des signaux analogiques que numériques. Dans le cadre de l'échantillonnage régulier, le filtrage est clairement défini depuis des années, permettant de définir des relations entre les échantillons d'entrée et de sortie selon deux grandes catégories : les filtres numériques à réponse impulsionnelle finie – RIF – et à réponse impulsionnelle infinie – RII. Or, dans le cadre d'un échantillonnage non uniforme, il n'y a plus de correspondance entre les instants d'échantillonnage et les indices des signaux numériques; le retard entre deux points consécutifs n'est plus constant; la transformée en Z n'est donc plus applicable pour concevoir l'équation aux différences. Pour réaliser un filtre RIF asynchrone dont l'équation aux différences est également le produit de convolution numérique entre le signal d'entrée et la réponse impulsionnelle, nous sommes alors revenus à la définition du produit de convolution à temps continu. Nous avons ainsi défini un nouveau produit de convolution à temps discret dit *asynchrone* : il consiste à calculer à partir de signaux échantillonnés non uniformément, un produit de convolution analogique à un instant donné en interpolant les signaux à temps discret, en les multipliant puis en calculant leur intégrale. Afin de ne pas expliciter l'interpolation, nous avons introduit un algorithme itératif qui décompose le résultat en une somme d'aires élémentaires obtenues directement à partir des échantillons. Nous avons étudié plusieurs produits de convolution différents en fonction de l'interpolation pour obtenir, à l'ordre 0, un filtre numérique RIF et une architecture itérative dont la complexité combinatoire moyenne est environ le double de l'architecture synchrone (Tableau 13). Ensuite, pour réaliser un filtre RII dont l'équation aux différences ne peut être ni conçue avec la transformée en Z, ni implémentable par un produit de convolution, est obtenue par la discrétisation du filtre analogique équivalent, représenté dans l'espace d'état. En rappelant les différentes

techniques de discrétisation dont certaines ont déjà été utilisées dans d'autres travaux de recherche, nous avons étudié la discrétisation en y intégrant des schémas évolués de type Runge-Kutta d'ordre 3 et d'ordre 4. Nous avons alors étudié leur stabilité vis-à-vis des paramètres du filtre et en avons déduit un flot de conception prenant en compte les spécificités de chaque schéma. Deux filtres numériques et leur architecture associée ont été définis en utilisant les schémas bilinéaire et de Runge-Kutta d'ordre 4 avec des caractéristiques différentes. Le premier est en effet inconditionnellement stable mais nécessite alors une inversion de matrice matériellement coûteuse. Le filtre est donc décomposé en structures simples du premier ou du deuxième ordre dont la complexité de traitement par point est environ le triple du cas synchrone. Le second filtre est en revanche stable sous certaines conditions (qui dépendent à la fois du filtre et des intervalles de temps du signal échantillonné) et présente l'avantage d'être implémentable dans une structure complète. Ce dernier est cinq fois plus complexe que l'implémentation synchrone (Tableau 13).

Enfin, une fois la chaîne de traitement entièrement considérée, nous avons établi un critère permettant d'évaluer numériquement, en fonction du signal, quelle technologie est à privilégier. Comme le traitement d'un échantillon est systématiquement plus complexe dans le cas asynchrone, l'unique solution pour que l'activité de la chaîne soit plus faible que dans le cas synchrone, est de traiter sur une durée égale, un nombre significativement plus petit de points. Nous proposons alors un modèle afin de représenter, pour un traitement donné, la complexité du calcul d'un point. Celui-ci dépend du niveau dans lequel on se place : nous considérons ici simplement le nombre total d'additions et de multiplications. Le gain de la technologie asynchrone est alors favorable (i.e. supérieur à un) si le rapport entre le nombre de points « synchrones » et « asynchrones » est supérieur au rapport des complexités de calcul. Malheureusement comme, dans les deux cas, le nombre de points dépend des paramètres d'échantillonnage, il n'est pas possible d'établir une correspondance entre l'activité économisée et l'activité du signal. Cependant, une extrapolation effectuée sur un signal de parole tend à montrer que pour un signal présentant une inactivité supérieure à 50% du temps total, le traitement effectué par une chaîne asynchrone permet de réduire la charge de calcul totale et donc la consommation électrique du circuit.

Traitement	*Filtre RIF*	*Filtre RII*	
Technique	Produit de convolution	Discrétisation de l'espace d'état analogique	
Méthode	Produit de convolution asynchrone d'ordre 0	Schéma numérique bilinéaire	Schéma numérique Runge-Kutta d'ordre 4
Stabilité	Inconditionnelle	Inconditionnelle	Conditionnelle
Fonctions réalisables	Toutes	Toutes	Toutes
Forme de la phase	≈ linéaire	Non linéaire	Non linéaire
Architecture	Itérative	Décomposée	Complète
N_{sync}/N_{async} **min**	2,7	3,5	5,25
Inactivité minimum	28%	16%	46%

Tableau 13 : Synthèse des caractéristiques des filtres étudiés dans ce travail de thèse

Ce travail, à la frontière du traitement du signal et de la microélectronique, ouvre de larges perspectives d'études. Du point de vue du traitement du signal, de nouvelles fonctions plus complexes sont à explorer, notamment au niveau de la régulation numérique. En effet, un système régulé et échantillonné régulièrement est « aveugle » entre deux occurrences. Or, si le système évolue significativement pendant ce temps là, le correcteur peut réagir trop tard voire même ne pas réagir du tout. Les systèmes sont donc naturellement suréchantillonnés entraînant une activité accrue et donc une augmentation de la consommation et des émissions électromagnétiques. Dans le domaine de l'industrie automobile par exemple, l'utilisation, dans certains équipements, d'un système régulé asynchrone, adapté sur la grandeur de sortie, permettrait de diminuer d'une part la puissance des émissions, alors que les normes imposent des contraintes de plus en plus drastiques, et de diminuer d'autre part la consommation permanente du véhicule. Ce type d'application semble particulièrement intéressant mais il faudra toutefois analyser les conditions de stabilité en boucle fermée d'un système échantillonné non uniformément. Parallèlement, un second aspect est à considérer : le traitement simultané de l'amplitude et de l'intervalle temps. En effet, actuellement

nous ne traitons que l'amplitude du signal d'entrée, l'intervalle de temps de sortie étant égal à l'intervalle de temps en entrée. Or, il serait peut-être judicieux d'étudier des traitements dont les couples de valeurs en sortie sont différents des couples en entrée. Bien que cela n'ait pas été encore étudié et semble être une perspective intéressante sur le plan théorique, elle pose immédiatement deux questions : quelle est l'opération à effectuer sur l'intervalle de temps sachant que la grandeur que l'on veut traiter à la base est l'amplitude du signal analogique ? Comment permettre un traitement des données en continu, si les intervalles de temps de sortie sont modifiés (reconstruction en temps réel d'un signal à temps continu) ?

Par ailleurs, du point de vue matériel, l'implémentation des blocs de traitement est à réaliser. Les modèles de complexité pourront alors être mis à jour avec une granularité plus faible qu'aujourd'hui, ce qui permettra d'affiner l'analyse du choix de la technologie à un plus bas niveau. Un Convertisseur Analogique/Numérique Asynchrone ayant été fabriqué en juin 2005, une plateforme d'expérimentation pourra ensuite être conçue afin de mesurer les performances d'une chaîne de traitement asynchrone en termes de consommation électrique, émissions électromagnétiques, rapidité… Bien que ce travail soit en cours de réalisation, il apparaît clairement que les bénéfices de cette approche seront exploitables pour la conception de circuits intégrés complexes ouvrant ainsi des perspectives nouvelles pour la réalisation de « capteurs intelligents » et de systèmes de communication à faible consommation et faibles émissions électromagnétiques.

Bibliographie

Abrial A., Bouvier J., Renaudin M., Senn P. et Vivet P.*"A New Contactless Smart Card IC using On-Chip Antenna and Asynchronous Microcontroller"*. IEEE Journal of Solid-State Circuits **36**(7): 1101-1107. 2001.

Aeschlimann F.*"Traitement Numérique du Signal sur des Signaux Echantillonnés Irrégulièrement"*. Rapport de DEA, Institut National Polytechnique de Grenoble. Juin 2002.

Aeschlimann F., Allier E., Fesquet L. et Renaudin M. *"Asynchronous FIR Filter: Towards a New Digital Processing Chain"*. Tenth IEEE International Symposium of Asynchronous and Systems, ASYNC'04. 198-206. Crête, Grèce. 19-23 Avril 2004.

Aeschlimann F., Allier E., Fesquet L. et Renaudin M. *"Analyse Spectrale de l'Echantillonnage par Traversée de Niveaux"*. 20e Colloque du Traitement du Signal et des Images, GRETSI. Louvain-la-Neuve, Belgique. 06-09 Septembre 2005.

Aeschlimann F., Allier E., Fesquet L. et Renaudin M. *"Spectral Analysis of Level-Crossing Sampling Scheme"*. International Workshop on Samping Theory and Applications, SAMPTA'05. Samsun, Turkey. Juillet 2005.

Aeschlimann F., Fesquet L., Allier E. et Renaudin M.*"Non-Uniform Sampling Scheme based on Level-Crossing of Periodic Signals"*. Sampling Theory on Signal and Image Processing - an International Journal, special issue on Non-Uniform Sampling: article soumis. 2005.

Allier E.*"Interface Analogique Numérique Asynchrone : Une Nouvelle Classe de Convertisseurs basés sur la Quantification du Temps"*. Thèse de doctorat, Institut National Polytechnique de Grenoble. Novembre 2003.

Allier E., Sicard G., Fesquet L. et Renaudin M. *"A New Class of Asynchronous Analog to Digital Converters"*. Third Workshop on Asynchronous Circuits and Systems Design, ACiD'03. Heraklion, Crete, Greece. 26-27 January 2003.

Allier E., Sicard G., Fesquet L. et Renaudin M.*"Asynchronous Level Crossing Analog to Digital Converters"*. Measurement Journal **37**(4): 296-309. 2005.

Bagshaw P. C. et Sarhadi M.*"Analysis of Samples of Wideband Signals Taken at Irregular, Sub-Nyquist, Intervals"*. IEE Electronics Letters **27**(14): 1228-1230. 1991.

Beutler F. J.*"Error-Free Recovery from Irregularly Spaced Samples"*. SIAM Review **8**(3): 328-335. 1966.

Bouesse F.*"Contribution à la conception de circuits intégrés sécurisés : l'alternative asynchrone"*. Thèse de doctorat, Institut National Polytechnique de Grenoble. Décembre 2005.

Clark W. A. *"Macromodular Computer Systems"*. AFIPS Conference Proceedings: 1967 Spring Joint Computer Conference. Atlantic City, NJ, Academic Press. **30:** 335--336. 1967.

Crouzeix M. et Mignot A. L.*"Analyse numérique des équations différentielles"*, Masson. 1992.

De Waele S. et Broersen P. M. T.*"Error Measures for Resampled Irregular Data"*. IEEE Trans. Instrumentation and Measurement **49**(2): 216-222. 2000.

Delmas J. P.*"Eléments de Théorie du Signal: les Signaux Déterministes"*. Paris, Editions Ellipses. 1991.

Duffin R. et Schaeffer A.*"A Class of Nonharmonic Fourier Series"*. Trans. Amer. Math. Soc. **72**: 341-366. 1952.

Duijndam A. J. W. et Schonewille M. A.*"Nonuniform Fast Fourier Transform"*. Geophysics **64**(2): 539-551. 1999.

Dutt A. et Rokhlin V.*"Fast Fourier Transforms for Nonequispaced Data"*. SIAM Journal of Scientific Computing **14**(6): 1368-1393. 1993.

Eldar Y. C. et Oppenheim A. V.*"Filterbank Reconstruction of Bandlimited Signals from Nonuniform and Generalized Samples"*. IEEE Transactions on Signal Processing **48**(10): 2864-2875. 2000.

Feichtinger H. G. et Gröchenig K. *"Theory and Practice of Irregular Sampling"*. in Wavelets: Mathematics and Applications. Boca Raton - Florida, CRC Press, Studies in Advanced Mathematics**:** 305-363. 1994.

Fontaine L.*"Traitement des Signaux à Echantillonnage Irrégulier. Application au Suivi Temporel de Paramètres Cardiaques"*. Thèse de Doctorat, Institut National Polytechnique de Lorraine. 1999.

Fontaine L. et Ragot J.*"Filtrage de signaux à échantillonnage irrégulier"*. Traitement du signal **18**(2): 89-101. 2001.

Fontolliet P.-G.*"Systèmes de Télécommunications - Bases de Transmissions"*, Dunod. 1983.

Gear C. W.*"Numerical Initial Value Problems in Ordinary Differential Equations"*, Prentice Hall. 1971.

Greengard L. et Lee J. Y.*"Accelerating the Nonuniform Fast Fourier Transform"*. SIAM Review **46**(3): 443-454. 2004.

Hauck S.*"Asynchronous Design Methodologies: An Overview"*. Proceedings of the IEEE **83**(1): 69-93. 1995.

Huffman D. A. *"The Synthesis of Sequential Switching Circuits"*. Sequential Machines: Selected Papers. E. F. Moore, Addison-Wesley. 1964.

Jerri A. J.*"The Shannon Sampling Theorem - Its Various Extensions and Applications: A Tutorial Review"*. Proceedings of the IEEE **65**(11): 1565-1596. 1977.

Kadec M. I.*"The Exact Value of the Paley-Wiener Constant"*. Soviet. Math. Dokl. **5**: 559-561. 1964.

Kinniment D. J., Gao B., Yakovlev A. V. et Xia F. *"Towards Asynchronous A/D Conversion"*. Fourth IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC'98. 206-215. San Diego, CA. 1998.

Kinniment D. J. et Yakovlev A. V.*"Low Power, Low Noise Micropipelined Flash A/D Converter"*. IEE Proc. Circuits Devices Syst. **146**(5): 263-267. 1999.

Kinniment D. J., Yakovlev A. V. et Gao B.*"Synchronous and Asynchronous A/D Conversion"*. IEEE Transactions on Very Large Scale Integration Systems **8**(2): 217-220. 2000.

Landau H. J.*"Sampling, Data Transmission, and the Nyquist Rate"*. IEEE Proc. **55**: 1701-1706. 1967.

Levinson N.*"Gap and Density Theorems"*. New York, Coll. Publ. 26, Amer. Math. Soc. 1940.

Li Y. W., Shepard K. L. et Tsividis Y. *"Continuous-Time Digital Signal Processors"*. Eleventh IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC'05. 138-143. New York, USA. March 2005.

Mark J. W. et Todd T. D.*"A Nonuniform Sampling Approach to Data Compression"*. IEEE Transactions on Communications **COM-29**(1): 24-32. 1981.

Martin A. J. *"The Limitations to Delay-Insensitivity in Asynchronous Circuits"*. Advanced Research in VLSI. W. J. Dally, MIT Press**:** 263-278. 1990.

Martin A. J. *"Programming in VLSI: From Communicating Processes to Delay-Insensitive Circuits"*. Developments in Concurrency and Communication. C. A. R. Hoare, Ed. Addison-Wesley**:** 1-64. 1991.

Martin R. J.*"Irregularly Sampled Signals : Theories and Techniques for Analysis"*. Thèse de doctorat, University College, London. 1998.

Marvasti F. A. *"Recovery of Missing Samples using a Novel Iterative Method"*. International Workshop on Sampling Theory and Applications, SAMPTA. Samsun, Turkey. July 2005.

Miller R. E.*"Switching Theory: Sequential Circuits and Machines Vol. 2"*, Vol. 2, John Wiley & Sons. 1965.

Myers C. J.*"Asynchronous Circuit Design"*, John Wiley & Sons. 2001.

Nita L. C.*"Analyse Spectrale de Signaux Aléatoires à Temps Continu Echantillonnés Non Uniformément"*. Thèse de doctorat, Université Paris XI d'Orsay / Supélec. Mars 2000.

NuHag.*"Numerical Harmonic Analysis Group"*. Faculté de Mathématiques, Université de Vienne: *www.univie.ac.at/NuHAG*.

Oppenheim A. V., Willsky A. S. et Young I. T.*"Signals and Systems"*. New-Delhi, Prentice Hall Ed. 1995.

Paley R. E. A. C. et Wiener N.*"Fourier Transform in the Complex Domain"*. Amer. Math. Soc. Coll. Publ. **19**. 1934.

Panyasak D.*"Réduction des émissions électromagnétiques des circuits intégrés : l'alternative asynchrone"*. Thèse de doctorat, Institut National Polytechnique de Grenoble. Juin 2004.

Papoulis A.*"Generalized Sampling Expansion"*. IEEE Trans. Circ. Syst. **24**(11): 652-654. 1977.

Poulton D. et Oksman J.*"Filtrage de signaux à échantillonnage non uniforme"*. Revue du Traitement du signal **12**(2): 81-88. 2001.

Prado J.*"Filtres numériques - Synthèse"*. Techniques de l'Ingénieur **E3**: 160_1 - 160_29. 2000.

Rabiner L. R. et Schafer R. W.*"Digital Processing of Speech Signals"*, Prentice Hall Inc. 1978.

Renaudin M., Vivet P. et Robin F. *"ASPRO: an Asynchronous 16-bit RISC Microprocessor with DSP Capabilities"*. IEEE European Solid-State Circuit Conference, ESSCIRC'99. 428-431. Duisburg. 21-23 Septembre 1999.

Santos D., Ferreira P. et Vieira J. *"Study of the Recovery of Missing Samples for Function and Derivative Oversampled Filter Bank"*. International Workshop on Sampling Theory and Applications, SAMPTA. Samsun, Turkey. Juillet 2005.

Sayiner N., Sorensen H. V. et Viswanathan T. R.*"A Level Crossing Scheme for A/D Conversion"*. IEEE Transactions on Circuits ans Systems II **43**(4): 335-339. 1996.

Shapiro H. S. et Silverman R. A.*"Alias-Free Sampling of Random Noise"*. SIAM Journal on Applied Mathematics **8**(2): 225-248. 1960.

Sutherland I. E.*"Micropipelines"*. Communications of the ACM **32**(6): 720-738. 1989.

Tarczynski A. et Allay N.*"Spectral Analysis of Randomly Sampled Signals: Suppression of Aliasing and Sampled Jitter"*. IEEE Transactions on Signal Processing **52**(12): 3324-3334. 2004.

Tarczynski A. et Tzvetkov K. *"Evaluation of Several Random Sampling Schemes for DASP Applications"*. International Workshop on Sampling Theory and Applications, SAMPTA'05. Samsun, Turkey. Juillet 2005.

Tsividis Y. *"Digital Signal Processing in Continuous Time: a Possibility for Avoiding Aliasing and Reducing Quantization Noise"*. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'04. Vol. 2. 589-592. Montreal, Canada. Mai 2004.

Tsividis Y., Cowan G., Li Y. W. et Shepard K. L. *"Continuous-Time DSPs, Analog/Digital Computers and Other Mixed-Domain Circuits"*. IEEE European Solid-State Circuit Conference, ESSCIRC'05. 113-116. Grenoble, France. 12-16 Septembre 2005.

Udding J. T.*"A Formal Model for Defining and Classifying Delay-Insensitive Circuits and Systems"*. Distributed Computing **1**(4): 197--204. 1986.

Venkataramani R. et Bresler Y.*"Perfect Reconstruction Formulas and Bounds on Aliasing Error in Sub-Nyquist Nonuniform Sampling of Multiband Signals"*. IEEE Transactions on Information Theory **46**(6): 2173-2183. 2000.

Venkataramani R. et Bresler Y.*"Optimal Sub-Nyquist Nonuniform Sampling and Reconstruction for Multiband Signals"*. IEEE Transactions on Signal Processing **49**(10): 2301-2313. 2001.

Werther T.*"Reconstruction from Irregular Samples with Improved Locality"*. Rapport de Magister, Université de Vienne. 1999.

Wojtiuk J. J.*"Randomised Sampling for Radio Design"*. Thèse de doctorat, University of South Australia. 2000.

Yao K. et Thomas J. O.*"On Some Stability and Interpolatory Properties on Nonuniform Sampling Expansions"*. IEEE Trans. Circuit Theory **CT-14**(4): 404-408. 1967.

Résumé

Ce travail de thèse s'intègre dans le cadre du développement de nouvelles approches de conception afin de réduire significativement la consommation électrique des Systèmes sur Puce (SoC) ou des Objets Communicants utilisés pour traiter numériquement des signaux. Le but est alors d'obtenir des systèmes entièrement contrôlés par les événements contenus dans les signaux. Dans ce contexte, une nouvelle catégorie de chaîne de traitement est définie, associant une implémentation matérielle asynchrone (sans horloge globale) et un échantillonnage non uniforme dans le temps dit « par traversée de niveaux ». Un convertisseur Analogique/Numérique dédié à cette tâche ayant déjà été réalisé, ce travail se focalise sur le traitement des données composées de couples amplitude-temps dont cette thèse montre que toute opération doit obligatoirement prendre en compte l'information temporelle. Des filtres numériques à réponse impulsionnelle finie (RIF) et infinie (RII) sont alors définis dans le cadre de signaux échantillonnés non uniformément. Des architectures sont proposées puis comparées à celles utilisées classiquement montrant que la complexité combinatoire était accrue. Un critère sur le choix de la technologie à privilégier, spécifiant la charge de calcul totale sur une durée finie, montre alors qu'en diminuant le nombre de points traités, l'approche asynchrone peut compenser le surcoût de complexité. Ainsi le traitement de signaux faiblement actifs par une chaîne asynchrone, combinant échantillonnage non uniforme et conception asynchrone, permet de réduire son activité moyenne et donc la consommation du circuit intégré, rendant cette technologie très attractive pour le domaine des SoC.

Mots clés : circuits asynchrones, échantillonnage non uniforme par traversée de niveaux, schéma d'échantillonnage, filtre RIF, filtre RII, architecture.

"NON UNIFORMLY SAMPLED SIGNAL PROCESSING: ALGORITHM AND ARCHITECTURE"

Abstract

This PhD thesis deals with the development of new design approaches in order to reduce significantly the power consumption of Systems on Chips (SoC) and Communicating Objects used in digital signal processing. The goal is to obtain systems only driven by the events contained in the useful signal. In this context, a new kind of signal processing chain is proposed, combining an asynchronous design (no global clock) and a non-uniform sampling scheme called level-crossing sampling. As an analog-to-digital converter dedicated to this task has already been studied, this work is focussed on the sampled signal processing based on amplitude-time couples. A preliminary study shows that any operation has to use the temporal information. Then, Finite and Infinite Impulse Response Filters (FIR and IIR) are defined in the case of non-uniform sampled signals. Architectures are also proposed and compared to those commonly used showing that the computational complexity is increased. A criterion about the choice of the technology to favour, specifying the global computational load over a finite time, has proved that the asynchronous approach can compensate the complexity cost by decreasing the number of processed points. Thus the processing of low-active signals by an asynchronous chain, combining asynchronous design and non-uniform lead to a reduction of its average activity and so of the power consumption of the integrated circuits making this technology very attractive for Soc area.

Key words: asynchronous circuits, non-uniform level-crossing sampling, FIR filter, IIR filter, architectures

ISBN : 2-84813-078-4