POLYPUBLIE

Modélisation et caractérisation d'une architecture générique de systèmes ultrasonores portables

Nicolas-Joël Vaillancourt-Veilleux

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	2006
Type	Mémoire ou thèse / Dissertation or Thesis Mémoire de maîtrise/Master's thesis
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/7918/
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Vaillancourt-Veilleux, N.-J. (2006). Modélisation et caractérisation d'une architecture générique de systèmes ultrasonores portables [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7918/

Détails de la thèse
Thesis details

Directeurs de recherche Advisors	Mohamad Sawan
Programme Program	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MODÉLISATION ET CARACTÉRISATION D'UNE ARCHITECTURE GÉNÉRIQUE DE SYSTÈMES ULTRASONORES PORTABLES

NICOLAS-JOËL VAILLANCOURT-VEILLEUX

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE ÉLECTRIQUE)

MAI 2006

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-19339-6
Our file Notre référence
ISBN: 978-0-494-19339-6

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MODÉLISATION ET CARACTÉRISATION D'UNE ARCHITECTURE GÉNÉRIQUE DE SYSTÈMES ULTRASONORES PORTABLES

Présenté par : VAILLANCOURT-VEILLEUX Nicolas-Joël

En vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

A été dûment accepté par le jury d'examen constitué de :

M. BRAULT Jean-Jules, Ph.D., président de jury

M. SAWAN Mohamad, Ph.D., directeur et membre

M. BERTRAND Michel, Ph.D., membre

REMERCIEMENTS

L'aboutissement de ce travail n'aurait pas été possible sans le concours de plusieurs personnes que je tiens à remercier très sincèrement ici :

- Monsieur Mohamad Sawan, pour ses conseils avisés tout au long de la rédaction de ce mémoire, et pour avoir su me transmettre une partie de son savoir et de sa vision du monde biomédical.

- Mes parents pour leur appui moral indéfectible pendant toute la période de mes études.

- Un merci particulier à Mathieu Gendron de l'institut de génie biomédical pour la conception des fantômes. Merci au génie de nos spécialistes informatiques M. Vesey et à M. Lepage pour tous les dépannages informatiques et un grand merci à Robert Chebli pour ses conseils et son savoir tout au long de ma maîtrise.

- Le président de mon jury, M. Brault et l'un des membres, M. Bertrand

- Enfin, plus que des remerciements pour: M. Choquette, Mme Desrochers, Mme Greaves, M. Mathieu, le CIRCUS, l'AÉCSP, M. Désilets, M. Lebel, M. Ferron, M. Auclair, M. Lesbros, Mme Leroux, Mme. Lévesque, Mme Carrier, Mme Ouellet et tous les autres qui ont supporté sauts d'humeur, week-ends ensoleillés et pourtant studieux, et nuits blanches rythmées au son du clavier pendant ces derniers 6 mois et qui ont su me remotiver quand le besoin s'en faisait sentir, m'aider à ne penser qu'au mémoire... sans porter aucune attention particulière au houblon, aux conquêtes stratégiques ou à la philosophie de la vie et qui m'ont permis de garder un souvenir indestructible de mes aventures aux cycles supérieurs. Merci !!!

RÉSUMÉ

Avec l'émergence des MEMS (Micro Electro Mechanical Systems) et des procédés de fabrication de puces haut voltage (300V), il est maintenant possible de créer des systèmes ultrasonores portables entièrement embarqués. Le défi dans ce domaine technologique concerne la conception et la vérification de ces systèmes. Notre équipe de recherche a développé différents circuits constituant un système ultrasonore complet. Par contre, le système entier n'a jamais pu être testé. À la lumière de nos recherches, nous avons constaté l'absence de modèles complets de systèmes ultrasonores permettant le développement de systèmes embarqués. Les hypothèses liées au développement d'un tel modèle seraient de savoir s'il permet : d'accélérer la vitesse de vérification, de valider de nouveaux concepts, d'accroître la flexibilité des tests effectués et de faire l'optimisation de systèmes complets, ceci afin de réduire le temps et les coûts de développement.

L'objectif de ce mémoire est donc de créer un environnement de tests dans lequel évolue un modèle comportemental générique de systèmes ultrasonores. Le but du modèle est de recréer le fonctionnement complet de l'imagerie ultrasonore afin de pouvoir évaluer les étapes du traitement des signaux. Cette évaluation des signaux peut se faire sur un module ou sur l'ensemble du système et a pour objectif d'en optimiser la fonctionnalité. En ce qui concerne l'environnement de tests, il permet la vérification et la validation de tous les modules et de l'ensemble du système à partir de signaux réels ou numérisés. Puisque la programmation du modèle est générique et qu'elle est effectuée en Verilog et en Verilog-A, l'environnement admet aussi l'utilisation des homologues réels (circuits schématique et dessins des masques) des modules comportementaux. Les logiciels utilisés sont ModelSim, CADENCE IC tools et Matlab. La méthode de validation et de vérification de l'ensemble du projet est de type comparatif.

ABSTRACT

With the emergence of MEMS (Micro Electro Mechanical Systems) and process of fabrication of high voltage chips (300V), it is now possible to create ultrasonic systems entirely embedded. The challenge in this technological field is the conception and verification of those systems. Our research team developed different circuits, forming a complete ultrasonic system. On the other hand, the entire system was never tested. In light of our research, we noticed the non-existence of complete ultrasonic models allowing the development of embedded systems. The hypothesis linked to developing such a model would be to know if it allows: to accelerate the verification speed, to validate new concepts, to increase the flexibility of tests and to maximise the entire system. All this to reduce the duration and the costs of development.

Therefore, the objective of this memoire is to create a testing environment in which evolves a generic behavioural model of ultrasonic systems. The goal of the model is to recreate the entire operation of the ultrasonic imagery, to then be able to evaluate the steps of the signal processing. This treating of the signals can be done on a module or on the entire system and has the objective to maximise its functions. In regards to the testing environment, it allows the verification and validation of all modules and of the entire system through real or scanned signals. Because the system's programming is generic and done on Verilog and Verilog-A, the environment also allows the use of real homologues (schematised circuits and mask drawings) of behavioural models. The softwares used are ModelSim, CADENCE IC TOOLS and Matlab. The method of validation and verification of the entire project is comparative.

TABLE DES MATIÈRES

REMERCIEMENTS ..... **IV**

RÉSUMÉ ..... **V**

ABSTRACT ..... **VI**

TABLE DES MATIÈRES ..... **VII**

LISTE DES FIGURES ..... **X**

LISTE DES TABLEAUX ..... **XIV**

SIGLES ET ABRÉVIATIONS ..... **XV**

INTRODUCTION **1**

CHAPITRE 1 ..... **4**

TECHNIQUES D'IMAGERIE MÉDICALE **4**

1.1 LES APPAREILS D'IMAGERIE MÉDICALE ULTRASONORE ACTUELS 4

1.2 THÉORIE DES ULTRASONS 5

1.2.1 Générer une onde ultrasonore (Transmission) 10

1.2.2 Information que procure un écho ultrasonore (Réception) 12

1.3 CONCLUSION 17

CHAPITRE 2 ..... **18**

RÉCENTS DÉVELOPPEMENTS **18**

2.1 RÉCENTS DÉVELOPPEMENTS DES SYSTÈMES ULTRASONORES INTÉGRÉS 18

2.2 VUE D'ENSEMBLE DES TRAVAUX DE NOTRE ÉQUIPE DE RECHERCHE 22

2.3 REVUE DÉTAILLÉES DES TRAVAUX ANTÉRIEURS 25

2.3.1 Bloc analogique 25

2.3.2 Bloc numérique 27

2.4 BANC D'ESSAI ANALOGIQUE ET NUMÉRIQUE 31
2.5 OUTILS DE PROGRAMMATION VHDL – VERILOG – MATLAB 33
2.6 PROBLÉMATIQUES 34
2.7 CONCLUSION 36

CHAPITRE 3 37

ENVIRONNEMENT DE SIMULATION : MÉTHODOLOGIE ET CONCEPTION 37

3.1 TRAVAUX RÉALISÉS 38
3.2 ACQUISITION ET SEGMENTATION (VOIR ANNEXE, MODELSIM) 41
3.2.1 Fantômes 41
3.2.2 Acquisition 43
3.2.3 Segmentation 45
3.3 MODÈLE NUMÉRIQUE COMPORTEMENTAL (VOIR ANNEXE, MODELSIM) 47
3.4 AFFICHAGE (VOIR ANNEXE , MATLAB) 51
3.5 MODULES ANALOGIQUES DU SYSTÈME (VOIR ANNEXE, CADENCE) 53
3.5.1 La conversion numérique à analogique CNA 53
3.5.2 Modèle du TLA 54
3.6 ANALYSE DU BLOC NUMÉRIQUE 56
3.7 CONCLUSION 59

CHAPITRE 4 61

SYNTHÈSE DES SIMULATIONS ET DES RÉSULTATS 61

4.1 RÉSULTATS PROVENANT DU MODÈLE NUMÉRIQUE PROPOSÉ 61
4.2 RÉSULTATS DU MODULE D'AFFICHAGE 66
4.3 TRAITEMENT ANALOGIQUE ET NUMÉRIQUE DU SIGNAL 72
4.3.1 Simulation du modèle du TLA 74
4.3.2 Simulation du modèle du TGC 75
4.3.3 Traitement numérique des signaux ultrasonores 77
4.4 VALIDATION ET VÉRIFICATION DES CIRCUITS RÉALISÉS PAR NOTRE ÉQUIPE 80

4.5 AUTRES ASPECTS DES RÉSULTATS 84
4.6 AUTRE DÉVELOPPEMENT DU PROJET 86

CONCLUSION 88

RECOMMANDATIONS SUR LE PROJET 91

BIBLIOGRAPHIE 94

ANNEXE 101

MODELSIM 101
CADENCE 122
MATLAB 129

LISTE DES FIGURES

Figure 1-1 Schéma de la réflexion et de la réfraction en fonction d'une onde incidente [66] 8
Figure 1-2 Schéma des différents types de réflectivité : a) spéculaire et b) diffusée [45].. 9
Figure 1-3 Schéma de la conception d'une sonde standard : a) un transducteur [6] et b) une sonde mécanique complète [5]. 11
Figure 1-4 Schéma d'un faisceau ultrasonore montrant les régions de Fresnel et de Fraunhofer [8]. 13
Figure 1-5 Schéma d'une impulsion ultrasonore et de ses échos dans le temps..... 14
Figure 1-6 Schéma des milieux Z_1 et Z_2 produisant l'onde de la Figure 1-5..... 14
Figure 2-1 Sonde intégrée avec la technologie CMUT [4]. 19
Figure 2-2 Schéma bloc du « beamforming » [17]. 21
Figure 2-3 Schéma bloc du DSP [17]. 21
Figure 2-4 Système d'interpolation SST [12]. 22
Figure 2-5 Schéma du bloc de l'interpolation mixte (NNI et linéaire) [17]. 23
Figure 2-6 Représentation graphique de l'interpolation mixte [17]. 23
Figure 2-7 Schéma bloc de notre système ultrasonore portable avec un étage d'entrée représentant une seule fois l'architecture des huit modules d'entrée qui sont directement connectés au DBF. 24
Figure 2-8 Illustration de la caractéristique entrée-sortie de six étages à gain dual du TLA [8]. 26
Figure 2-9 Illustration de l'échelle de compensation du TGC [13]. 27
Figure 2-10 Illustration des délais variables utilisés pour le balayage électronique [17]. 29
Figure 2-11 Image utilisée pour les simulations de la partie numérique : a) image de départ et b) image en coordonnées polaires [17]. 30
Figure 2-12 Banc d'essai constitué d'un même stimulus pour différents modèles [24]... 32
Figure 2-13 Schéma bloc de l'implantation du PUIP [21]. 33

Figure 3-1 Schéma bloc de l'environnement de simulation d'un système ultrasonore. ... 38
Figure 3-2 Image des deux fantômes utilisés pour le projet : a) représente les inclusions à distance variable à 3% d'agar, b) les inclusions équidistantes à pourcentage d'agar variable ... 42
Figure 3-3 Image échographique d'une section du cœur obtenue avec le Scanview 1 : VG = Ventricule Gauche, OG = Oreillette Gauche, BiC = Bicuspides ... 42
Figure 3-4 Banc d'essai de l'acquisition de données échographiques avec un fantôme : A) échographe, B) sonde, C) analyseur et D) fantôme. ... 43
Figure 3-5 Les trois signaux de synchronisation a) écho, b) « start scan » et ... 46
Figure 3-6 Schéma bloc des modules du modèle numérique comportemental ... 48
Figure 3-7 Schéma bloc de la reconstruction du signal «Original » analogique. ... 54
Figure 3-8 Circuit schématique d'un étage à gain dual du TLA [8]. ... 54
Figure 3-9 Diagramme bloc de l'amplificateur TGC à gain programmable [8]. ... 56
Figure 3-10 Schéma bloc du système numérique implanté en VHDL : a) le « Beamforming » et b) le DSP et le « Scanconverter ». ... 57
Figure 3-11 Schéma bloc du « Scanconverter » : a) calcul de la position xy et b) calcul de l'interpolation ... 58
Figure 3-12 Résultat obtenu par notre équipe avec la figure 1.10a [17] ... 59
Figure 4-1 Image 1 provenant directement de l'appareil de Scanview. ... 62
Figure 4-2 Graphique temporel des échos provenant des lignes de balayage 65 et 66 pour une profondeur de 5cm. ... 63
Figure 4-3 Graphique temporel des échos provenant des lignes de balayage 65 et 66 pour une profondeur de 10cm. ... 63
Figure 4-4 Graphique de l'enveloppe IQ du signal de la ligne de balayage 65 pour 5cm.64
Figure 4-5 Graphique de l'enveloppe simple du signal de la ligne de balayage 65 pour 5cm ... 64
Figure 4-6 Signal de l'enveloppe simple agrandi. ... 66
Figure 4-7 Signal de l'enveloppe IQ agrandi ... 66

Figure 4-8 Image homogène : a) sans interpolation, b) interpolation linéaire, c) NNI et d) interpolation mixte. 68
Figure 4-9 Résultat image 1 réel, obtenu avec le modèle et sans modification. 68
Figure 4-10 Image 1 : a) sans interpolation, b) avec interpolation mixte, c) avec interpolation linéaire et d) avec NNI. 69
Figure 4-11 Image 1 avec un filtre simple et une interpolation mixte. 70
Figure 4-12 Image 1 obtenue après une compression en log à base 2. 71
Figure 4-13 Image numérisée du fœtus produite par notre modèle : a) sans interpolation, b) sélection du pixel le plus intense, c) la moyenne de 2 et d) le dernier pixel est gardé. 72
Figure 4-14 Signal « Original » après sa reconstitution et son filtrage pour une profondeur de 10cm. 73
Figure 4-15 Fonction de transfert du modèle comportemental du TLA. 74
Figure 4-16 Signal « Original » après l'amplification du TLA. 74
Figure 4-17 Résultat du TLA lorsque son gain est diminué. 75
Figure 4-18 Fonction de transfert du modèle comportemental du TGC en fonction du temps pour des paramètres de programmation maximal. 76
Figure 4-19 Signal « Original » amplifié par le modèle du TGC avec les paramètres de gain maximaux pour une profondeur de 10cm. 76
Figure 4-20 Signal « Original » amplifié par le modèle du TGC avec les paramètres moyens pour une profondeur de 10cm. 77
Figure 4-21 Graphique de l'enveloppe IQ du signal « Original » pour 5cm et amplifié par le TGC moyen. 78
Figure 4-22 Graphique de l'enveloppe IQ du signal « Original » pour 5cm et amplifié par le TGC maximum. 78
Figure 4-23 Graphique de l'enveloppe IQ du signal « Original » pour 10cm et amplifié par le TGC maximum. 79
Figure 4-24 Résultat obtenu avec le signal « Original » et le TGC moyen pour une profondeur de 10cm. 79

Figure 4-25 Résultat du circuit schématique du TLA..... 81
Figure 4-26 Signal « Original » amplifié par le circuit schématique du TGC avec les paramètres moyen pour une profondeur de 10cm..... 82
Figure 4-27 Image 1 de test pour le module numérique de l'équipe: a) Image utilisée pour la simulation et b) image résultant de la simulation. 83
Figure 4-28 Image 2 obtenue avec a) Scanview et b) le modèle 83
Figure 4-30 Image du cœur obtenu par la modélisation. 84
Figure 4-31 Architecture des combinaisons de simulation possibles 86

LISTE DES TABLEAUX

Tableau 1-1 Comparaison entre différents systèmes ultrasonores sur le marché 6
Tableau 1-2 Propriétés de différents matériaux biologiques (célérité, impédance, atténuation et dépendance fréquentielle) [38] 7
Tableau 4-1 Gain programmer des champs du TGC. 75
Tableau 4-2 Temps de simulation de l'environnement de tests 85

SIGLES ET ABRÉVIATIONS

Ampli-op (Amplificateur Opérationnel)

ASIC (Application Specific Integrated Circuits)

CAN (Convertisseur Analogique Numérique)

cm (Centimètre)

CMC (Canadien Microelectronic Center)

CMOS (Complementary Metal-Oxide Semiconductor)

CMUT (Capacitive Micromachined Ultrasonic Transducer)

CNA (Convertisseur Numérique – Analogique)

CT (Computer tomography)

dB (Décibel)

DBF (Digital Beamforming)

DSC (Digital Signal Converter)

DSP (Digital Signal Processing)

FIR (Finite Impulse Response)

FPGA (Field Programmable Gate Array)

Front-End (Étage d'entrée)

HDL (High Device Langage)

HV (High Voltage)

IIR (Infinite Impulse Response)

IRM (Imagerie par Résonnance Magnétique)

IUS (Imagerie par Ultrason)

LCD (Liquid Cristal Display)

LKT (Look up Table)

MEMS (Micro Electromecanichal Systems)

mV (Millivolt)

NNI (Nearest Neighbor Interpolation)

PUIP (Programmable Ultrasound Image Processor)

SNR (Signal to Noise Ratio)

SST (Selective Sampling Technic)

TGC (Time Gain Compensator)

TLA (True Logarithmic Amplifier)

µm (Micromètre)

µs (Microseconde)

µV (Microvolt)

VHDL (Very High Speed Integrated Circuit Hardware Description Language)

VP (Video Processing)

Introduction

La détection d'un cancer, la vérification du sexe et de la santé physique d'un fœtus, le débit sanguin ou encore les malformations d'organes sont des exemples de diagnostics médicaux qui requièrent des méthodes d'imagerie interne du corps humain. De nos jours, les milieux hospitaliers sont équipés de plusieurs appareils d'imagerie médicale qui utilisent différentes technologies comme l'imagerie par résonance magnétique, l'imagerie par rayons X et l'imagerie par ultrasons. Les systèmes ultrasonores ont des avantages qui méritent une attention particulière. Par contre, ce champ technologique est très vaste. C'est la raison pour laquelle nous abordons une branche spécifique de l'imagerie médicale, soit les systèmes ultrasonores portables. Nos recherches portent plus précisément sur l'élaboration d'un modèle de système ultrasonore pouvant être intégré sur une ou plusieurs puces. La pertinence de nos recherches est de tenir compte des contraintes liées au développement de systèmes ultrasonores sur puce afin d'arriver à la conception d'un prototype viable. L'objet de cette maîtrise est donc de concevoir un modèle et un environnement de tests pouvant valider et vérifier les différents modules que nécessite la conception d'un système ultrasonore.

Le premier chapitre fait un survol des techniques d'imagerie médicale existant afin de bien saisir le contexte actuel de cette technologie. La branche dans laquelle nous avons concentré nos recherches est présentée en fournissant un aperçu des tendances du développement des systèmes d'échographie ultrasonores sur le marché. Finalement, toute la théorie concernant les ultrasons est exposée dans la dernière section. Ceci afin de faire ressortir les paramètres connus qui doivent être pris en compte dans la conception d'un banc d'essai qui permet l'élaboration d'expériences à variables contrôlées et à groupes de contrôle.

Le chapitre deux présente les récents développements dans le domaine des systèmes ultrasonores portables. La première section montre que les recherches en matière de miniaturisation rendent viable la conception d'un système ultrasonore portable. La seconde section dévoile le concept du système ultrasonore portable proposé par notre équipe de recherche. La troisième section détaille les travaux antérieurs effectués par notre équipe de recherche puisque l'un des objectifs est d'en faire la caractérisation et la validation. Les faits indiquent qu'un projet nécessite en temps 30 % de conception et 70 % de vérification et de validation. Nous présentons donc, à la section 4, deux techniques de tests employés : l'une est employée dans l'industrie et l'autre a été développée récemment. L'objectif est de montrer la nécessité de faire la conception d'un banc d'essai efficace pour tester un système complexe miniaturisé qui utilise un ensemble de modules analogiques et numériques. Par la suite, les outils utilisés pour la modélisation sont présentés dans le but de montrer l'approche de validation et de vérification qui est employée. Ce premier chapitre se termine finalement par la description des hypothèses de travail qui sont présentées sous forme de problématiques.

Le chapitre trois décrit la conception de l'environnement de simulation. Cette section débute avec la description de l'environnement de tests proposé lors de ce projet de maîtrise afin de voir l'ampleur des travaux effectués. La section 2 présente la conception de l'environnement d'acquisition des données ultrasonores. Elle comprend le montage, les appareils employés, la fabrication des fantômes et la provenance des images de tests. La seconde partie présente toutes les données en rapport avec la segmentation du signal acquis afin de pouvoir l'utiliser lors des simulations. Par la suite, la conception de chaque module du modèle global du système ultrasonore est présentée. Ceci comprend toutes les parties du modèle comportemental numérique et analogique ainsi que l'affichage. De plus, certains modules possèdent plus d'une technique de fonctionnement. La raison est que le projet doit justement prouver qu'il est possible de faire varier facilement le comportement des modules pour pouvoir déceler l'impact sur le système en entier. Enfin, la dernière section présente l'analyse complète des lacunes du système numérique

proposé par l'équipe et les endroits où des améliorations doivent être apportées pour rendre le système fonctionnel.

Le chapitre quatre présente la synthèse des simulations et des résultats. La première section fournit les résultats de tests prouvant la fonctionnalité du banc d'essai pour tous les groupes de modules : numérique, affichage et analogique. De plus, la section analogique inclut les résultats des simulations du système en entier. La seconde section est consacrée à la vérification et à la validation des modules générés par l'équipe de recherche. Enfin, la dernière section présente les temps de simulation et un graphique de la méthodologie de tests employée.

Finalement, une analyse générale du projet est présentée afin de démontrer si toutes les hypothèses de départ ont été atteintes. Nous établissons aussi les améliorations du projet comme conclusion de ce mémoire.

CHAPITRE 1

Techniques d'imagerie médicale

L'objectif visé par ce chapitre est de situer notre projet dans le contexte actuel et d'illustrer les techniques entourant l'imagerie médicale. Ainsi, nous présentons en première partie les différents appareils à ultrasons, les conventionnels et les portables, employés sur le marché. La seconde partie la théorie entourant les ondes ultrasonores pour permettre le traitement efficace des données qui seront traitées par le système modélisé.

1.1 Les appareils d'imagerie médicale ultrasonore actuels

Les appareils ultrasonores ont une architecture de système commune. Cette architecture se divise en quatre parties :

1) La sonde permet de générer l'onde ultrasonore et sert aussi de récepteur pour les échos.
2) L'amplificateur analogique permet d'augmenter l'amplitude des échos selon différents paramètres.
3) Le traitement des signaux analogiques et numériques afin de pouvoir former une image.
4) La conversion des données traitées pour en faire l'affichage.

Par contre, les produits que l'on retrouve sur le marché sont parfois très différents. Le Tableau 1-1 présente une comparaison entre différentes compagnies qui produisent des appareils de diagnostic général. Le domaine des systèmes ultrasonores est si vaste que ce tableau ne présente que quelques compagnies se spécialisant dans le domaine du

diagnostic général. L'avant-dernière colonne présente des appareils de type conventionnel, soit des systèmes entièrement basés sur des ordinateurs de bureau qui possèdent une carte électronique spécialisée afin de pouvoir y brancher des sondes échographiques. Le traitement des données est alors entièrement fait de façon logicielle. Par contre, ces appareils de très haute performance sont lourds, encombrants et très difficiles à opérer [14]. La première colonne présente les appareils de type portable, à ne pas confondre avec les appareils de type mobile qui ne sont, en général, que des appareils conventionnels montés sur roue. Le type portable se distingue par le fait qu'il est léger, de très faible volume et consomme peu d'énergie. L'évolution des systèmes portables se divise en deux catégories très distinctes. La première catégorie a gardé le principe des appareils conventionnels. Par contre, elle utilise maintenant des ordinateurs portables. De plus, l'innovation technologique est au niveau de la carte électronique spécialisée utilisée comme interface avec la sonde. Elle est soit intégrée [33] ou simplement placée en petit module externe [65]. Ce type de systèmes utilise majoritairement des logiciels afin de traiter l'information. La deuxième catégorie a plutôt opté pour un traitement axé sur le traitement matériel [58] et sur le développement de systèmes sur puces [14] [64]. Les données du tableau montrent bien que les systèmes qui ont pris la voie du développement de puces dédiées (Application Specific Integrated Circuit – ASIC) et des processeurs dédiés sont beaucoup plus petits que les autres systèmes, tout en offrant autant de fonctionnalité [64].

1.2 Théorie des ultrasons

Les ultrasons sont des ondes mécaniques de hautes fréquences. Ils font partie du domaine des ondes acoustiques qui ont la propriété de se propager uniquement dans des milieux non vide. Le spectre de fréquence acoustique se divise en quatre parties : les infrasons qui ont une fréquence d'oscillation inférieure à 20 Hz, les sons audibles qui oscillent entre 20 Hz et 20 KHz, les ultrasons qui oscillent entre 20 KHz et 1 GHz et les hypersons qui ont

une fréquence supérieure à 1GHz. Dans le domaine du diagnostic médical, la propagation d'une onde ultrasonore se fait par une succession de pressions et de dépressions longitudinales. La vitesse de propagation ou la célérité de l'onde ultrasonore est dictée globalement par l'équation **(1.1)** et s'exprime en mètres par seconde [9] :

$$c = \frac{1}{\sqrt{\rho_0 K}} \qquad \textbf{(1.1)}$$

Tableau 1-1 Comparaison entre différents systèmes ultrasonores sur le marché

Compagnie	Appareil portable	Poids portable (Kg)	ASIC	Doppler	Volume du portable (cm)	Traitement Logiciel	Appareil conventionnel	Référence
Biosound Esaote	Falco	-	non	non	ordinateur portable	oui	Mylab50	[40]
Chison	Chison 600M	7	oui	non	42 x 27 x 25	oui	Chison 600A pro	[42]
Fukuda Denshi	Ffsonic UF-4100	10	non	non	33,2 x 31,2 x 26	oui	UF-850XDT	[44]
GE healthcare	Logiq book XP	5	oui	oui	ordinateur portable	oui	Logiq 9	[46]
Hitachi	EUB-405 plus	-	non	non	ordinateur portable	oui	HI Vision 8500	[47]
Kontron	Soneo	-	non	non	ordinateur portable	oui	Sigma 5000	[49]
Medison America	Mysono	4	non	non	ordinateur portable	non	Sonoace 8000 live	[54]
Philips medical	SD 100E system	10,5	non	non	ordinateur portable	oui	ie33	[55]
Shimadzu medical	SDU-350 XL	-	non	non	ordinateur portable	oui	SDU-2200 xplus	[58]
Siemens	ACUSON cypress	8	non	oui	ordinateur portable	oui	ACUSON sequoia	[59]
Sonosite	Sonosite 180plus	2,6	oui	oui	33 x 19,3 x 6,35	non	-	[62]
Terason (Teratech)	Terason t3000	4	oui	oui	ordinateur portable	oui	-	[63]
Toshiba	Famio 5	-	non	non	ordinateur portable	oui	Aplio XV	[65]
Ultrasonix medical	-	-	non	-	-	-	Sonix SP	[66]
Zonare Medical system	z.one	2,5	non	oui	ordinateur portable	oui	z.one	[69]
SIUI	CTS-485	-	non	non	40,5 x 29,5 x 26	non	Apogee 3500	[60]
Scanview	Scanview1	1,3	non	non	19,5 x 11,4 x 6,8	non	-	[56]

Le Tableau 1-2 présente la célérité de différents milieux. Cette variation de la célérité a un impact sur les caractéristiques intrinsèques de l'onde ultrasonore.

Tableau 1-2 Propriétés de différents matériaux biologiques (célérité, impédance, atténuation et dépendance fréquentielle) [38]

Material	Propagation speed, c ($m\ s^{-1}$)	Characteristic impedance, Z ($10^6\ kg\ m^{-2}\ s^{-1}$)	Attenuation coefficient, α at 1 MHz ($dB\ cm^{-1}$)	Frequency dependence of α
Air	330	0.0004	1.2	f^2
Blood	1570	1.61	0.2	$f^{1.3}$
Brain	1540	1.58	0.9	f
Fat	1450	1.38	0.6	f
Liver	1550	1.65	0.9	f
Muscle	1590	1.70	1.5–3.5	f
Skull bone	4000	7.80	13	f^2
Soft tissue (mean values)	1540	1.63	0.6	f
Water	1480	1.48	0.002	f^2

Comme le montre l'équation **(1.2)**, la longueur d'onde λ est directement reliée à la vitesse de propagation c et à la fréquence f de l'onde [9] :

$$\lambda = \frac{c}{f} \qquad \textbf{(1.2)}$$

Par ailleurs, le comportement d'un milieu matériel par rapport aux ultrasons est exprimé par une constante appelée impédance acoustique, Z (équation **(1.3)**. L'interface entre deux milieux de Z différents est exprimée par ΔZ, et ce traduit par l'aptitude d'un milieu (Z_1) à faciliter la pénétration des ondes ultrasonores provenant d'un autre milieu (Z_2). Le Tableau 1-2 montre quelques exemples pour différents milieux :

$$Z = \sqrt{\rho_0 / K} \text{ ou } Z = \rho_0 c \left(kg / m^2 \sec \right) \qquad \textbf{(1.3)}$$

Cette impédance vient jouer un rôle prédominant lorsqu'il est question de propagation dans le corps humain. Comme nous l'avons montré précédemment, l'anatomie humaine est constituée de différentes couches de constituants biologiques qui ont des impédances acoustiques diverses. Ce qui donne lieu à l'apparition d'interfaces entre les différents milieux, donc entre les différentes impédances acoustiques. Selon la loi de Snell, une onde qui pénètre une interface entre deux milieux d'impédances différentes occasionne deux phénomènes : la réflexion et la réfraction. À la Figure 1-1, en assumant que l'angle de l'onde incidente (θ_i) est égal à l'angle de l'onde réfléchie (θ_r), on peut déterminer l'angle de l'onde réfractée (θ_{rf}) avec l'équation **(1.4)**. Une fois θ_{rf} connu, il est possible, avec l'équation **(1.5)**, de déterminer le coefficient de réflectivité de la surface en fonction de l'angle incident [9].

$$\frac{\sin\theta_i}{\sin\theta_t} = \frac{\lambda_1}{\lambda_2} = \frac{c_1}{c_2} \qquad \textbf{(1.4)}$$

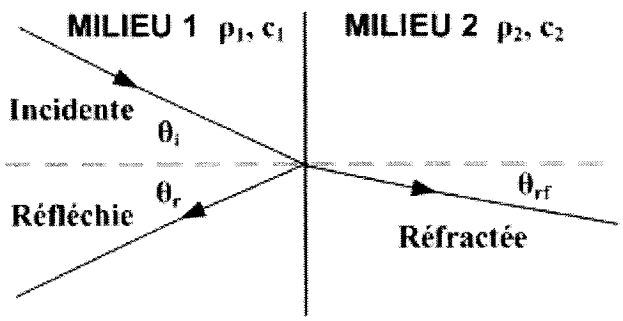


Figure 1-1 Schéma de la réflexion et de la réfraction en fonction d'une onde incidente [66] .

$$R = \frac{p_r}{p_i} = \frac{\dfrac{Z_2}{\cos\theta_{rf}} - \dfrac{Z_1}{\cos\theta_i}}{\dfrac{Z_2}{\cos\theta_{rf}} + \dfrac{Z_1}{\cos\theta_i}} \text{ si } \theta_i \neq \theta_{rf} \text{ ou } R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \text{ si } \theta_i = \theta_{rf} \qquad \textbf{(1.5)}$$

Où p_r est la pression réfléchie et p_i la pression incidente. À noter qu'il existe deux types de réflexion : spéculaire et diffusée. La réflectivité spéculaire, Figure 1-2a, se fait lorsque la sonde envoie une onde ultrasonore sur une grande surface lisse. Il en résulte une

grande dépendance à l'angle de réflexion. L'onde obéit alors parfaitement aux lois citées ci-dessus. Par contre, la réflexion par diffusion, Figure 1-2b, se fait sur de petits objets représentant les irrégularités d'une surface. Elle est beaucoup moins dépendante de l'angle de réflexion et moins intense. En somme, la propagation d'une onde ultrasonore dans le corps peut être affectée par la réflectivité, la réfractivité et la diffusivité des interfaces du milieu.

a)

b)

Figure 1-2 Schéma des différents types de réflectivité : a) spéculaire et b) diffusée [45].

L'absorption est un autre facteur qui affecte l'onde ultrasonore. Elle représente la variation de l'intensité acoustique de l'onde en fonction de sa pénétration dans le milieu, comme le montre l'équation **(1.6)** [9].

$$I_m = \frac{P^2}{\rho c} \qquad (\mathrm{W/cm^2}) \qquad \textbf{(1.6)}$$

Où I_m est l'intensité de l'onde et P est l'amplitude (valeur crête maximale) de la pression mécanique. L'intensité acoustique I_m se mesure en watts par unité de surfaces (W/cm^2) tandis que la puissance acoustique I_a se mesure en watts par unité de temps (W/min). Dans le corps, l'intensité ultrasonore (I_m) ne doit pas dépasser 100 mW/cm^2 [45]. En général, on utilise des valeurs d'intensité variant entre 10 μW et 10 mW/cm^2. L'ensemble des facteurs atténuant l'intensité de l'onde ultrasonore lors de sa propagation dans un

milieu est appelée atténuation. L'équation **(1.7)** permet de calculer la valeur de l'intensité après avoir parcouru une distance quelconque dans un milieu :

$$I_{mx} = I_{m0}\, e^{-2\alpha x} \quad (W/cm^2) \qquad \textbf{(1.7)}$$

Où I_{mx} est l'intensité à la distance x, I_0 est la valeur initiale de l'intensité de l'onde, α est le coefficient d'atténuation et x est la profondeur. Il faut noter que le coefficient d'atténuation tient compte de l'absorption et de la diffusion. Le Tableau 1-2 montre que le coefficient d'atténuation α augmente de façon quasi proportionnelle à la fréquence ou au carré de la fréquence selon le milieu de propagation.

1.2.1 *Générer une onde ultrasonore (Transmission)*

L'émission des ultrasons par une sonde ce fait de façon pulsée. Il y a deux modes possibles qui sont caractérisés par trois facteurs : continu et séquentiel [8] :

1) La fréquence de répétition des impulsions (Pulse Repetition Frequency - PRF).
2) Le rapport cyclique, qui est le rapport entre la durée d'une impulsion et le temps de répétition d'une impulsion.
3) La puissance acoustique I_a (W/min). Cette puissance est assujettie aux normes qui réglementent la durée et l'intensité des impulsions.

En fait, ces paramètres contrôlent la génération d'une onde ultrasonore.

En imagerie médicale, les ondes ultrasonores sont produites par des éléments piézoélectriques. Ces éléments ont la propriété de se déformer lors de l'application d'une différence de potentiel à leur surface. Ainsi, la déformation d'un élément piézoélectrique peut générer une onde mécanique (onde acoustique). L'équation **(1.8)** montre la relation entre la longueur de l'impulsion électrique (L_{imp}) et la fréquence de résonance de l'élément piézoélectrique (f_0). L'amplitude typique d'une impulsion électrique varie entre 50 et 200 volts.

$$L_{imp} = \frac{1}{2f} \qquad \textbf{(1.8)}$$

En général, les éléments piézoélectriques, aussi appelés transducteurs, sont des céramiques (PZT – titanate-zirconate de plomb), des polymères (PVDF – polyfluorure de vinylidène) ou des composites (PZT/polymère 1-3). Par contre, depuis quelques années, il y a eu de nouveaux développements dans la conception de membranes de transducteur sur des circuits intégrés. Il s'agit de films de matériaux piézoélectriques que l'on dépose sur des surfaces de silicium. Le CMUT qui est présenté à la section 2.1 est un bon exemple de cette nouvelle technologie. Il est donc certain que les appareils médicaux actuels générant des ondes ultrasonores, soit les sondes, changeront d'apparence dans un avenir proche.

Les sondes mécaniques actuelles, Figure 1-3b, ont toutes les mêmes constituants de base : le transducteur (ultrasound transducer), le mécanisme de balayage (ultrasound motor, optical sensor), un dôme (plastic dome), un fluide d'agencement (matching fluid) et un boîtier (plastic case). La Figure 1-3a présente un transducteur standard. Ce dernier est constitué d'un boîtier (casing), d'un piézoélectrique (polymer film) et de l'amortisseur (backing). L'amortisseur est placé derrière le transducteur. Il sert à atténuer fortement la résonance du piézoélectrique après son excitation afin de réduire la durée de l'impulsion ultrasonore.

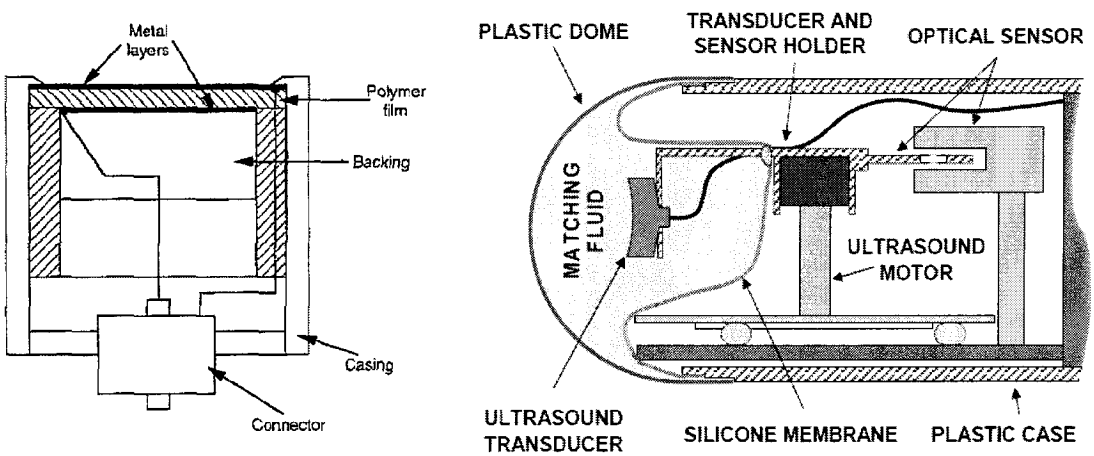


Figure 1-3 Schéma de la conception d'une sonde standard : a) un transducteur [6] et b) une sonde mécanique complète [5].

De plus, il empêche la propagation et la réflexion de l'onde produite sur la surface arrière du transducteur. Ainsi l'onde se propage uniquement dans un sens, mais elle perd légèrement de son efficacité et affecte directement le facteur de qualité (Q). Le dôme et le fluide servent à l'agencement des impédances acoustiques entre le transducteur et l'objet sondé. Car l'onde ultrasonore produite par l'élément piézoélectrique doit traverser deux interfaces : fluide-dôme (Z_1-Z_2) et dôme-milieu extérieur (Z_2-Z_3). De plus, le dôme peut, dans certains cas, servir de lentille et permettre la focalisation du faisceau ultrasonore, mais en général la lentille est directement placée sur le transducteur. Sa forme peut être soit concave ou convexe selon l'agencement acoustique employé. La focalisation peut aussi se faire en donnant une courbure convexe au transducteur.

Le faisceau d'ondes ultrasonores produit est présenté à la Figure 1-4 et se divise en deux sections appelées zone de Fresnel pour la région assez proche et zone de Fraunhofer en ce qui concerne la région éloignée. La longueur de la zone de Fresnel (L) est constituée d'ondes ultrasonores rectilignes, et comme le montre l'équation **(1.9)**, elle dépend du diamètre du transducteur ($\boldsymbol{D}$) et de la longueur d'onde des ultrasons ($\boldsymbol{\lambda}$). C'est à l'intérieur de cette zone que s'effectue la focalisation du faisceau ultrasonore. La zone de Fraunhofer est la partie où l'onde ultrasonore diverge. L'angle de divergence ($\boldsymbol{\theta}$) est régi approximativement par l'équation **(1.10)**. Maintenant que les caractéristiques des ondes ultrasonores sont connues, il faut connaître leurs impacts sur l'imagerie médicale par ultrasons :

$$L = D^2 / 4\lambda \qquad \textbf{(1.9)}$$

$$\theta = \sin^{-1}\left[1.22\lambda / D\right] \qquad \textbf{(1.10)}$$

1.2.2 *Information que procure un écho ultrasonore (Réception)*

La partie précédente présentait les propriétés de la transmission ou de la génération d'ondes ultrasonores. Par ailleurs, le transducteur sert aussi de récepteur. Si une onde de pression vient déformer le transducteur alors une différence de tension apparaîtra à ses

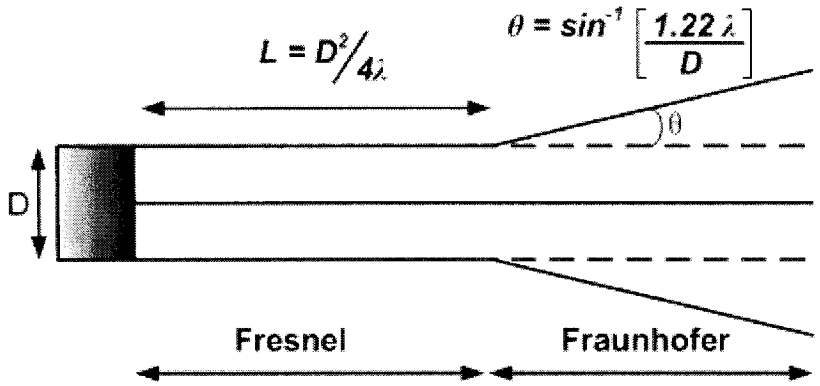


Figure 1-4 Schéma d'un faisceau ultrasonore montrant les régions de Fresnel et de Fraunhofer [8].

bornes. La Figure 1-5 présente un exemple de réception avec un transducteur. L'impulsion initiale correspond au type d'onde acoustique généré par la sonde. Un écho est produit lorsqu'il y a une réflexion de l'onde sur une interface. De plus, cette onde doit revenir vers la source afin d'être détectée lorsque la technique de détection par réflexion est utilisée. La Figure 1-6 montre le lien entre les interfaces traversées et le signal de la Figure 1-5. En effet, l'impulsion initiale correspond à une première réflexion entre la sonde et le milieu Z_1. Les deux autres réflexions sont dues aux interfaces Z_1-Z_2 (l'encadré) et Z_2-Z_1 (l'encerclement). Il est facile de remarquer qu'il y a beaucoup d'atténuation en comparant l'amplitude de l'impulsion initiale à celles des échos. Donc l'amplitude permet d'identifier une interface mais il faut aussi la localiser. Pour ce faire, il faut tenir compte du temps. L'écho est détecté uniquement après avoir fait un aller-retour, soit 2 fois le temps pour atteindre l'interface. Sachant que la vitesse de propagation d'un ultrason dans un milieu est constante alors on détermine la distance entre la sonde et l'interface par l'équation **(1.11)** :

$$\text{Distance} = \frac{c * Temps}{2} \qquad \textbf{(1.11)}$$

Cette représentation, présentée à la Figure 1-5, est appelée mode A. Il s'agit du signal analogique $S_{echo}(t)$ dont la représentation mathématique est donnée par l'équation **(1.12)** :

$$S_{echo}(t) = A(t)\cos(2\pi f_0 t + \varphi) \qquad \textbf{(1.12)}$$

Où $S_{echo}(t)$ est le signal ultrasonore, $A(t)$ la variation de l'amplitude et $cos(2\pi f_0 t)$ l'onde porteuse.

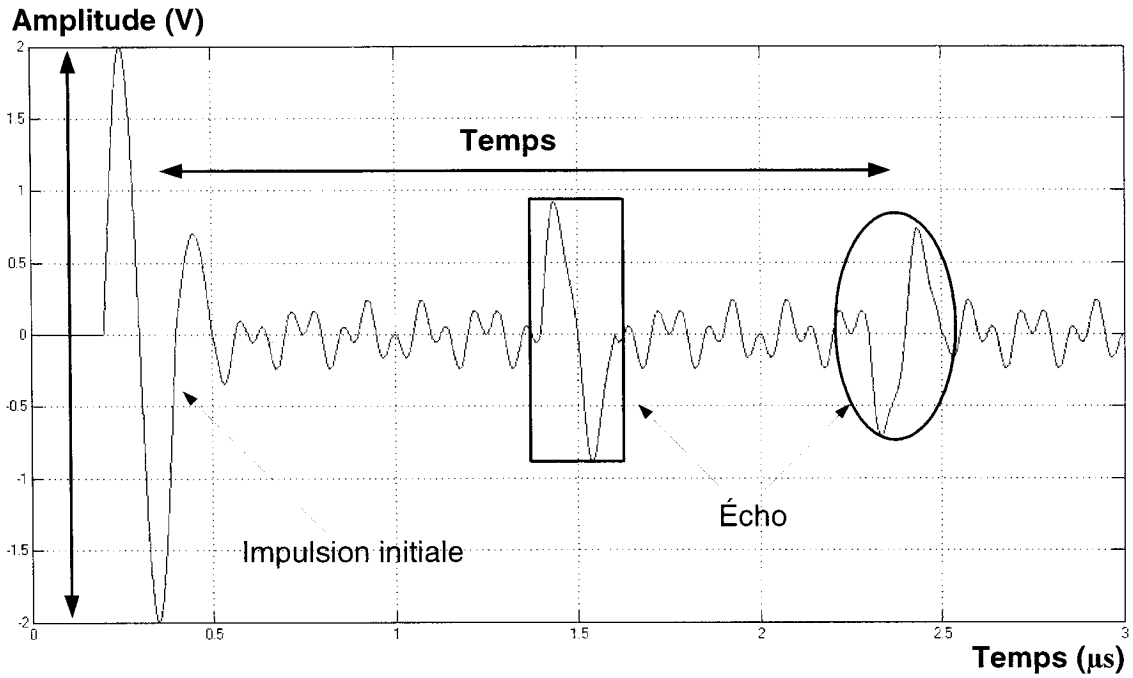


Figure 1-5 Schéma d'une impulsion ultrasonore et de ses échos dans le temps.

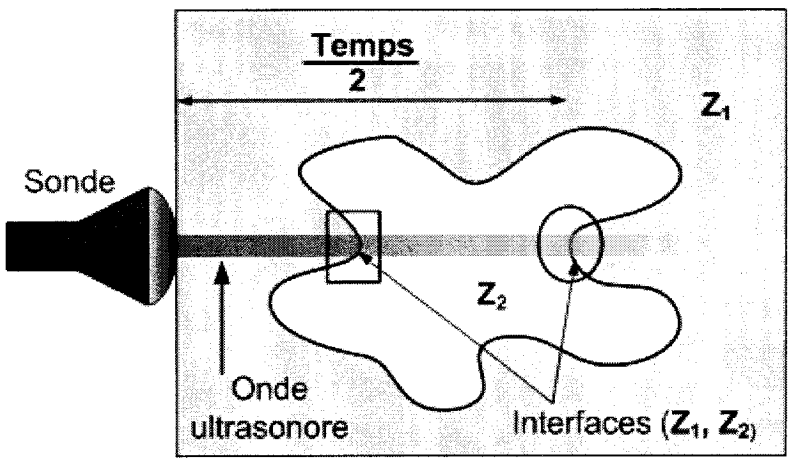


Figure 1-6 Schéma des milieux Z_1 et Z_2 produisant l'onde de la Figure 1-5.

Afin de produire une image sur un écran, il faut utiliser le mode B. Cette représentation des échos transforme l'amplitude des ondes en intensité de pixels. Ceci caractérise la plupart des systèmes d'imagerie ultrasonores qui utilisent différents tons de gris pour l'affichage. Enfin le mode M indique simplement que l'affichage est en temps réel.

La formation d'une image en deux dimensions ne peut pas se faire avec une seule onde ultrasonore dirigée dans une seule direction. Il est nécessaire de faire un balayage complet de la zone à visualiser. Cette technique consiste à couvrir une zone à l'aide de plusieurs faisceaux ultrasonores. Un exemple est présenté à la Figure 2-6. Le balayage effectué par la sonde peut être exécuté de façon mécanique ou électronique. La différence entre les deux mécanismes sera montrée ultérieurement.

Enfin, le dernier point important est la résolution que peut fournir une sonde ultrasonore. Il y a deux types de résolution qui caractérisent les systèmes ultrasonores : axiale et latérale. La résolution axiale (R_A), dans le sens de propagation de l'onde, se définit comme étant la distance minimale séparant deux interfaces rapprochées dont les échos respectifs ne se recouvrent pas. Elle dépend de deux éléments : le facteur de qualité Q du transducteur qui influence la longueur réelle de l'onde ultrasonore générée, et la longueur d'onde. Cette dernière dépend elle-même de la fréquence et de la célérité du milieu. La résolution latérale (R_L), perpendiculaire au sens de propagation de l'onde, se définit comme étant la distance minimale entre deux interfaces distinctes sur un même plan, de telle sorte qu'elles ne peuvent pas être confondues. Cette distance minimale est égale à la largeur du faisceau ultrasonore à cet endroit. Elle dépend de plusieurs éléments : la longueur d'onde, la longueur de la zone de Fresnel et l'angle de divergence. Les paramètres recueillis nous permettent alors de préparer un banc d'essais pour tester correctement un système ultrasonore et pour des fins d'optimisation. On peut donc maintenant tirer des conclusions sur tous les paramètres vus précédemment.

Paramètres liés à l'environnement :

1) La compressibilité du milieu K ;
2) La densité du milieu ρ_0 ;
3) La célérité c car elle dépend de K et ρ_0 ;
4) L'impédance acoustique Z car elle dépend de K et ρ_0 ;
5) La réfraction, car elle dépend de Z ;
6) L'absorption qui varie en fonction de c ;
7) La diffusion dépend de la surface de l'interface et du milieu ;
8) Le coefficient d'atténuation, qui dépend de la réfraction, de l'absorption et de la diffusion.

Paramètres liés à l'appareil :

1) La fréquence de l'onde ultrasonore f ;
2) La taille du transducteur D ;
3) Longueur d'impulsion L_{imp} ;
4) L'intensité de l'impulsion I_m ;
5) La longueur d'onde λ ;
6) La profondeur de balayage x qui dépend de I_{mx} ;
7) Le facteur de qualité Q qui dépend du transducteur utilisé ;
8) La résolution axiale qui dépend de Q et de λ ;
9) La résolution latérale qui dépend de λ et de la longueur de la zone de Fresnel.

Le signal électrique provenant des échos sera donc caractérisé par une fréquence, une amplitude, une résolution maximale axiale et latérale et un temps de propagation indiquant une profondeur, ce qui représente une seule dimension. L'imagerie en deux dimensions tient compte des mêmes paramètres en plus d'y ajouter la résolution d'image provenant de la technique de balayage employée et du traitement vidéo effectué.

1.3 Conclusion

Ce chapitre a présenté, en première partie, l'utilité de faire de l'imagerie médicale. Le survol de la constitution des tissus humains a permis de définir le milieu de propagation de l'onde ultrasonore. Aussi, la présentation des appareils utilisant différents principes physiques a servi à la définition des avantages et des inconvénients de l'IUS. De plus, il a été possible de faire ressortir les points forts de l'IUS afin de les optimiser ultérieurement. La section suivante a porté sur le développement des appareils actuels afin d'avoir une vue d'ensemble sur l'évolution des progrès dans le domaine des systèmes médicaux ultrasonores portables. Enfin, la dernière section a présenté les connaissances techniques reliées aux ultrasons, nécessaires à la conception de l'environnement de simulation, ceci afin de posséder tous les outils permettant une compréhension rigoureuse et optimale des tests à faire et des résultats à obtenir.

CHAPITRE 2

Récents développements

Ce chapitre présente les différents éléments de recherche qui ont servi de base à l'élaboration et au développement de ce projet. La première section concerne les faits saillants d'actualité sur l'avancement de la recherche dans la conception de systèmes échographiques portables en présentant les technologies existant sur le marché et dans la littérature. Dans la deuxième section, une revue détaillée des travaux antérieurs de notre équipe s'impose car ces travaux font partie intégrante du design complet de ce projet. La section trois fait un survol des méthodes de simulations employées et des outils logiciels qui sont utilisés pour la modélisation. Enfin nous introduisons la problématique qui nous a amené à notre projet, soit la modélisation et la caractérisation d'une architecture générique de systèmes ultrasonores portables.

2.1 Récents développements des systèmes ultrasonores intégrés

L'élaboration de systèmes sur puce comportant des éléments analogiques, numériques et différents types de mémoires (statique et dynamique) est rendue très complexe [2], surtout lorsqu'il est question de systèmes ultrasonores qu'il faut miniaturiser. Cette section permet d'avoir une vue d'ensemble des progrès effectués sur chaque élément de base constituant les systèmes ultrasonores : la sonde, l'amplificateur analogique, le traitement de signal et la conversion des données.

Jusqu'à maintenant la sonde n'avait pas beaucoup évolué du point de vue physique. En effet, les différents appareils utilisent tous des cristaux piézoélectriques et ne diffèrent que par la méthode d'utilisation et le nombre d'éléments utilisés [8]. La Figure 2-1

montre que Wygant et al. [40] ont proposé récemment une sonde intégrée sur puce, fabriqué avec la technologie CMUT (capacitive micromachined ultrasonic transducer). Au lieu d'utiliser des éléments piézoélectriques, ils utilisent des micro-membranes qui sont activées par des tensions de 30 volts. Ces membranes peuvent mesurer jusqu'à 150 um x 150 um, ce qui représente une surface de 0.0225 mm^2 qui est environ 1000 fois plus petite que celle d'un élément piézoélectrique conventionnel. De plus, comme il est possible de le voir sur la Figure 2-1, les mêmes auteurs présentent un étage d'amplification analogique intégré. Ils proposent une architecture constituée d'un préamplificateur à faible bruit (amplificateur de trans-impédance), d'un TGC normal et d'un convertisseur analogique - numérique (CAN) à plusieurs canaux [19]. Les avantages de ce type de système sont qu'il améliore le rapport signal sur bruit, réduit le nombre de connexions entre la sonde et un module externe de traitement de signal et fournit les moyens de relier électroniquement une matrice de plusieurs membranes.

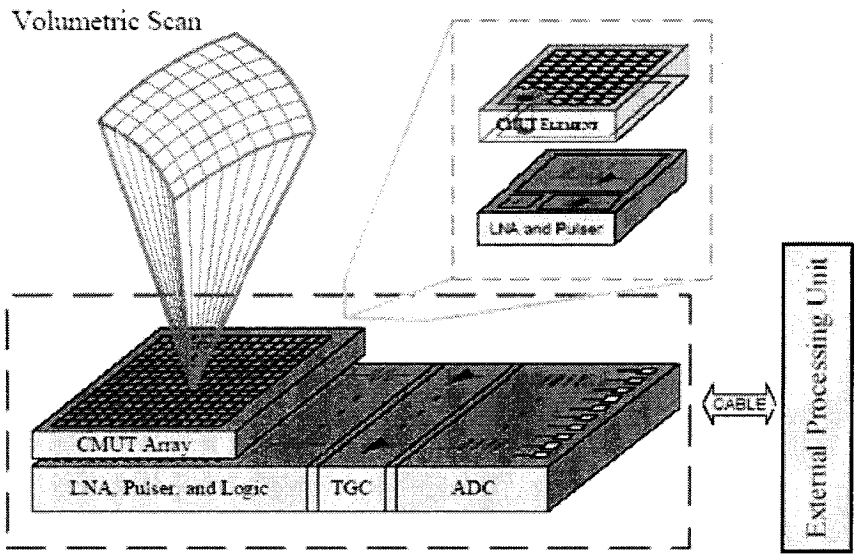


Figure 2-1 Sonde intégrée avec la technologie CMUT [4].

Notre équipe proposait récemment un étage d'amplification analogique constitué d'un amplificateur logarithmique (True Logarithmic Amplifier - TLA), d'un amplificateur temporel (TGC) programmable et d'un convertisseur analogique - numérique (CAN)

[34]. Le TLA a pour but d'amplifier de beaucoup les faibles signaux et faiblement les grands signaux. Ceci permet l'utilisation efficace de la plage dynamique de tension. Le TGC permet d'amplifier le signal en fonction de la profondeur, ce qui permet d'avoir un écho qui dépend presque uniquement de la surface réfléchie. Selon le type d'examen effectué (œil, abdomen, foie) la fréquence et la profondeur varient. Le TGC doit alors s'adapter, ce qui explique le fait qu'il soit programmable. Le CAN permet de convertir immédiatement les données en numérique afin de pouvoir les transmettre en réduisant considérablement le bruit.

Précédemment, il a été mentionné que la plupart des systèmes ultrasonores emploient des logiciels afin de réaliser le traitement des signaux numériques. Le défi est donc de produire un circuit électronique capable d'effectuer ces mêmes tâches. Plus récemment, notre équipe a proposé une architecture de traitement de signal qui se compose de deux parties : la mise en forme numérique du faisceau (DBF - Digital Beamforming) et le traitement numérique des signaux (DSP - Digital Signal Processing) [17]. Le DBF, présenté à la Figure 2-2, possède l'avantage de pouvoir effectuer une focalisation électronique dynamique lors de la réception de l'onde. Il utilise aussi peu de mémoire et il emploie un contrôleur dédié. Le DSP, aussi appelé démodulateur IQ (Figure 2-3), permet de faire la détection d'enveloppe et de réduire considérablement le bruit. L'ensemble des modules peut être implanté dans un circuit programmable (Field Programmable Gate Array – FPGA). Le modèle structurel du système est en langage VHDL. Il a été implanté dans un FPGA et testé sur une plateforme ARM-RRP[17].

La dernière partie du système ultrasonore qu'il faut convertir au niveau matériel est la conversion des données pour l'affichage. L'une des innovations apportées est un système mis au point par Fristsch et al. [12]. La Figure 2-4b montre une architecture appelée SST (Selective Sampling Technique) qui permet de faire la conversion des données angulaires, obtenue par le balayage ultrasonore, en coordonnées cartésiennes, ceci afin de pouvoir afficher les données sur un écran LCD. Les auteurs proposent un système

d'interpolation par région afin d'augmenter la définition de l'image, réduire les artéfacts et le temps de calcul. La Figure 2-4a montre le résultat de la conversion par interpolation SST qui emploie plus de lignes d'interpolation selon la profondeur de la zone balayée. Par exemple, la région k=1 est constituée uniquement de lignes de balayage, tandis que la

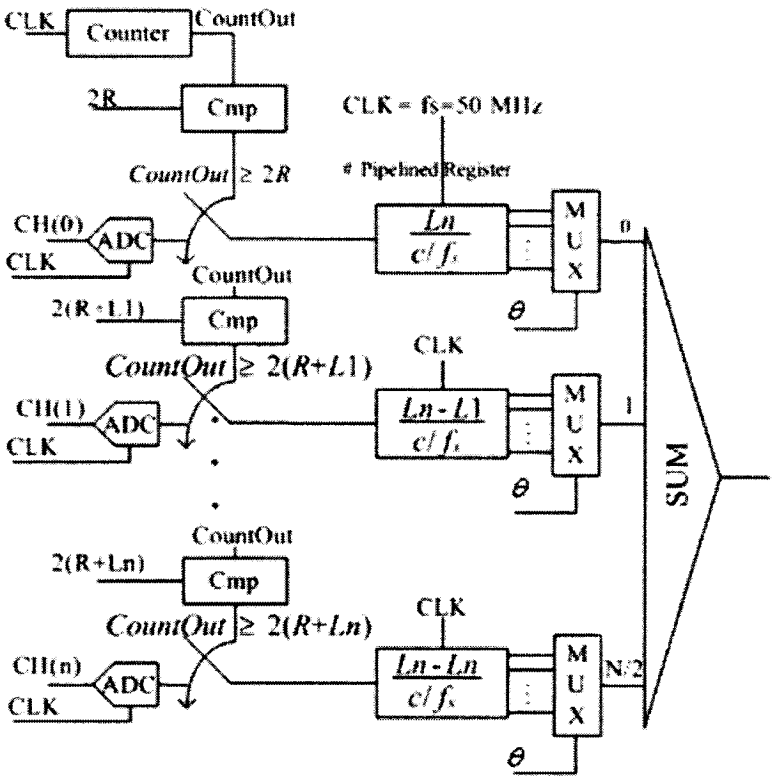


Figure 2-2 Schéma bloc du « beamforming » [17].

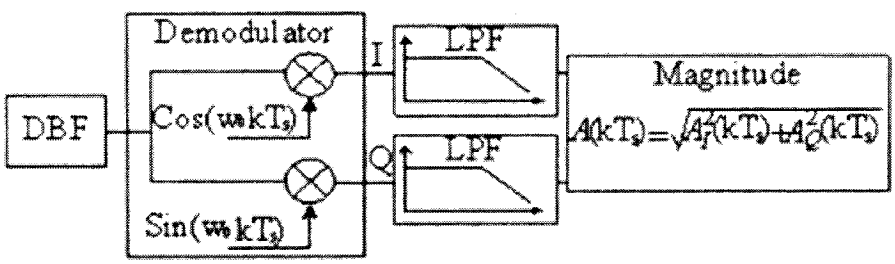


Figure 2-3 Schéma bloc du DSP [17].

région k=3 compte 3 lignes d'interpolation en plus. En comparaison avec d'autres systèmes de conversion, la SST a un bon rapport signal sur bruit (Signal to Noise Ratio - SNR), nécessite un nombre constant et modéré d'échantillons et utilise une arithmétique simple (Figure 2-4b).

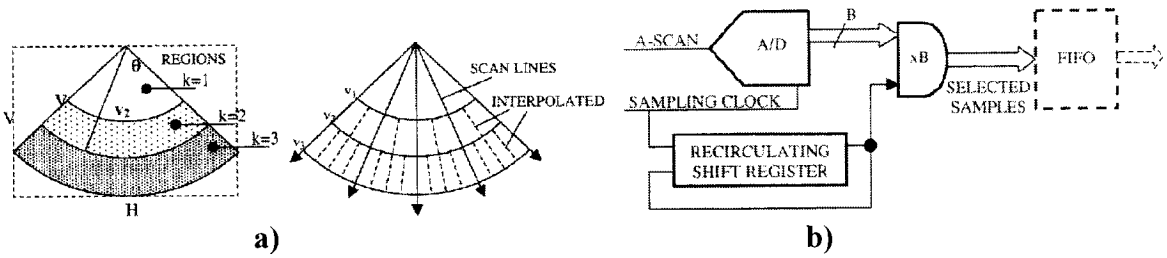


Figure 2-4 Système d'interpolation SST [12].

La Figure 2-5 présente le schéma bloc du système de conversion qui a récemment été mis au point par notre équipe Polystim [17]. Le système de conversion proposé est une combinaison de la méthode d'interpolation du plus proche voisin (NNI - Nearest Neighbor Interpolation) et la méthode d'interpolation linéaire, appelée méthode mixte. Elle a comme avantage d'augmenter la définition de l'image, de diminuer les artéfacts et de nécessiter peu de mémoire. La Figure 2-6 montre la représentation graphique de la méthode mixte. La NNI est une méthode d'interpolation qui consiste à transmettre la valeur d'un pixel au pixel voisin. Ainsi, à la Figure 2-6, il est possible de créer la ligne SL11 à partir de SL1 et SL12 avec SL2. L'interpolation linéaire est une autre méthode qui produit une nouvelle ligne, mais il faut deux lignes déjà existantes. Ainsi SL13 est une interpolation linéaire de SL11 et SL12.

2.2 Vue d'ensemble des travaux de notre équipe de recherche

Le projet de notre équipe de recherche est de développer un système ultrasonore portable. Les travaux des dernières années ont porté sur différents aspects des systèmes ultrasonores. En 1999, Beauchamp-Parent [3] développait un système discret d'alarme

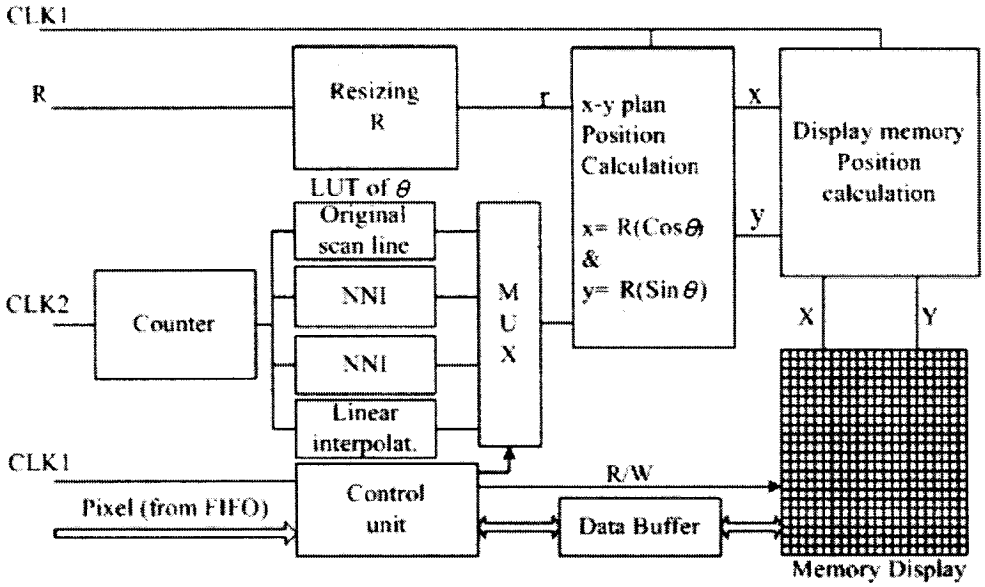


Figure 2-5 Schéma du bloc de l'interpolation mixte (NNI et linéaire) [17].

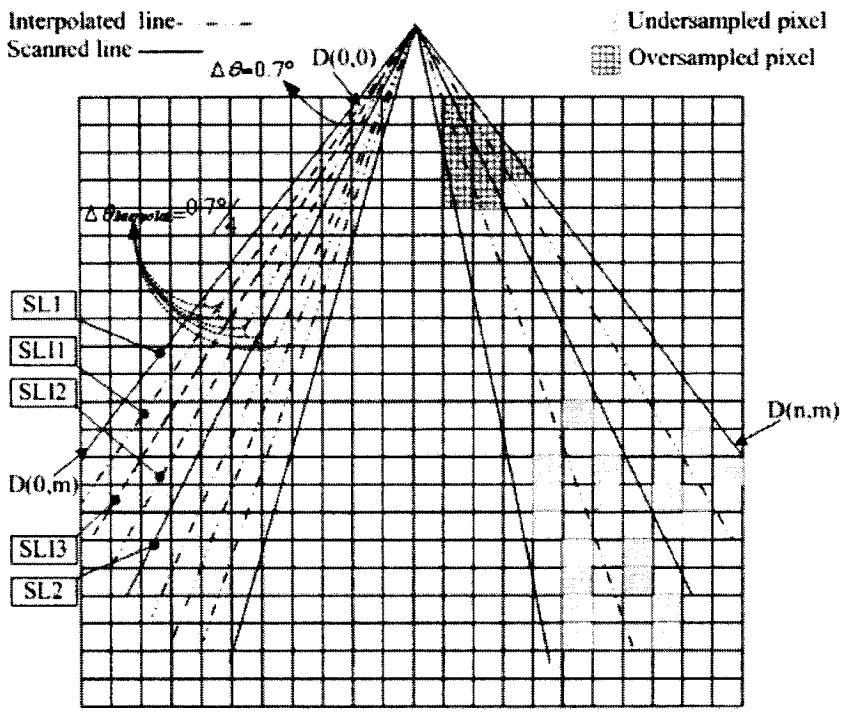


Figure 2-6 Représentation graphique de l'interpolation mixte [17].

de conditionnement ultrasonore programmable pour le traitement de l'énurésie. En 2003. Chebli [8] réalisait un étage de réception intégré d'un échographe portable. En 2005, Kassem [17] présentait ses travaux à propos d'une technique de conception système sur une puce dédiée à l'imagerie par ultrasons. Ce qui a mené au développement d'une architecture de système ultrasonore sur puce (Figure 2-7).

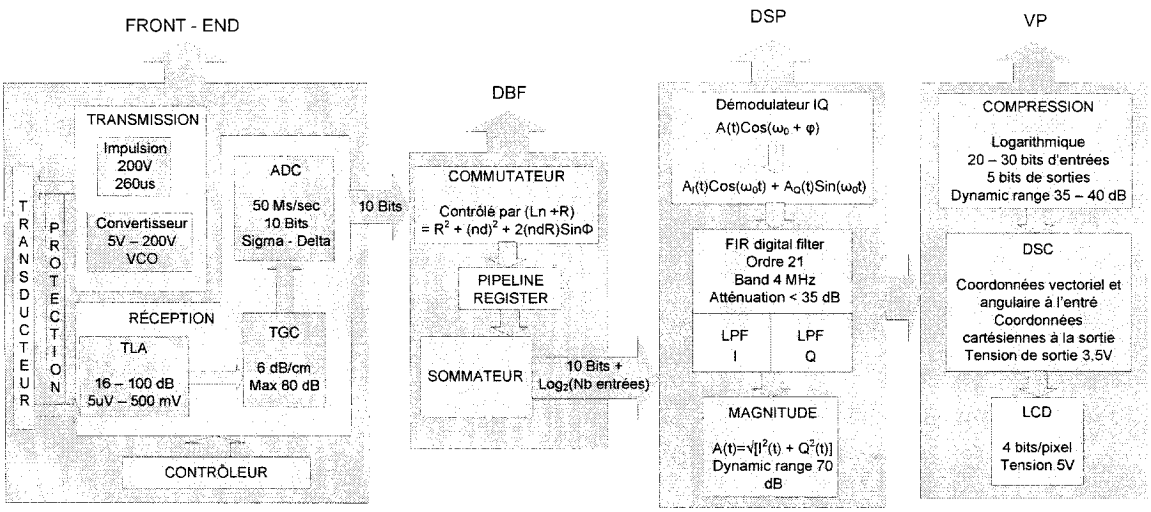


Figure 2-7 Schéma bloc de notre système ultrasonore portable avec un étage d'entrée représentant une seule fois l'architecture des huit modules d'entrée qui sont directement connectés au DBF.

Le concept de base est présenté à la Figure 2-7 et le fonctionnement du système peut être résumé comme suit : le transducteur (sonde) fait la réception de l'onde ultrasonore et fait la conversion en signal électrique. Le signal est ensuite transmis à l'étage d'entrée « Front-End » (amplificateur analogique) qui est constitué de 2 blocs analogiques : TLA et TGC et qui est répété 8 fois afin d'avoir 8 canaux de réception. Par la suite, le signal est numérisé par le CAN et dirigé vers le bloc numérique constitué du DBF et du DSP. Enfin l'image est convertie dans un format de données affichable sur un écran LCD par le bloc VP (Video Processing).

2.3 Revue détaillées des travaux antérieurs

2.3.1 *Bloc analogique*

Le système ultrasonore proposé par notre équipe de recherche, Figure 2-7, est constitué d'une partie analogique, le « front-end », qui contient le TLA, le TGC et l'ADC. Le TLA permet d'amplifier les signaux des échos reçus afin de normaliser l'intervalle de sortie. Le circuit schématique est composé de six blocs de gain dual cascadés un à la suite de l'autre. Le terme dual signifie que chaque bloc est composé de deux amplificateurs en parallèle. Un des amplificateurs est unitaire et l'autre est de gain A. La sortie d'un bloc est égale à la somme des valeurs des deux amplificateurs internes. Lorsqu'une tension V_{in} est présentée à l'entrée d'un bloc, elle est comparée à une tension de référence qui détermine si la tension doit être amplifiée d'un gain A ou simplement additionnée d'une valeur limite V_{oL}. Ainsi, il est possible de mettre ce comportement sous la forme des équations suivantes :

Le gain unitaire : $V_{out} = V_{in}$ **(2.1)**

Le gain limiteur :

$$V_{out} = A * V_{in} \quad \text{si } V_{in} < V_{ref} \qquad \textbf{(2.2)}$$
$$V_{out} = V_{oL} \quad \text{si } V_{in} \geq V_{ref}$$

Le gain du bloc :

$$V_{out} = (A + 1) V_{in} \quad \text{si } V_{in} < V_{ref} \qquad \textbf{(2.3)}$$
$$V_{out} = V_{oL} + V_{in} \quad \text{si } V_{in} \geq V_{ref}$$

La simulation du modèle schématique de la succession des six blocs permet d'obtenir la courbe présentée à la Figure 2-8, où chaque cercle délimite l'intervalle relié à un bloc. La manipulation des équations de gain de chaque bloc permet d'obtenir l'équation **(2.4)** qui caractérise cette courbe [8]. Les autres simulations servent à déterminer les principales caractéristiques du TLA, soit la largeur de bande, le bruit d'entrée, l'erreur logarithmique, etc. Les mêmes simulations ont été exécutées sous forme de tests sur la puce fabriquée à partir du modèle du dessin des masques :

$$V_o = \left(6 + \frac{1}{A} + \log_{A+1}\left(\frac{A * V_{in}}{V_{oL}}\right)\right) V_{oL} \qquad \textbf{(2.4)}$$

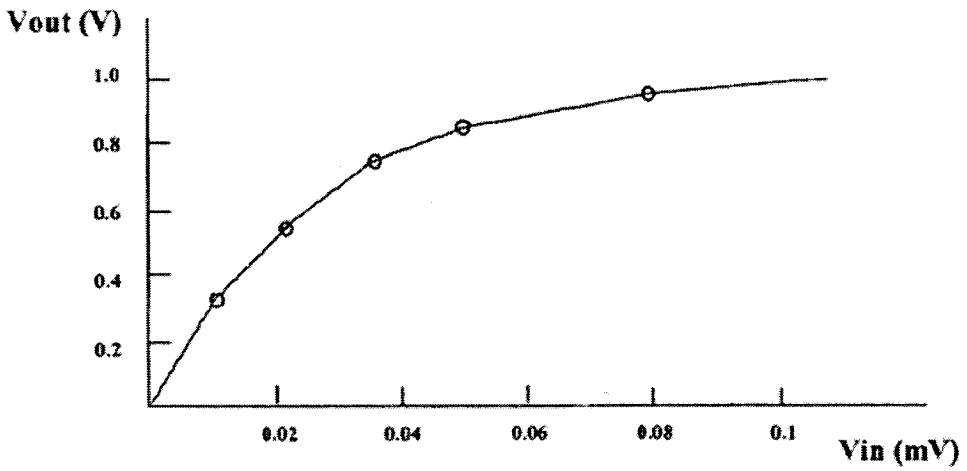


Figure 2-8 Illustration de la caractéristique entrée-sortie de six étages à gain dual du TLA [8].

Le TGC compense l'effet d'atténuation des échos dans les tissus afin que les signaux dépendent uniquement de la surface interceptée. Dans un système ultrasonore, les unités de temps sont remplacées par des unités de profondeur en utilisant l'équation **(1.11)**. Ceci permet de faire le lien entre le rôle du TGC et l'atténuation qui est en fonction de la profondeur (équation **(1.7)**). Le TGC programmable est constitué de quatre blocs cascadés à gain programmable possédant tous un gain maximal de 20dB. Ceci permet d'avoir une plage dynamique totale de 80dB. La programmation du gain est spécifique selon trois champs de profondeur comme le montre la Figure 2-9.

La première section n'est pas un champ, elle correspond à l'initialisation de la valeur du gain du TGC qui peut varier entre 0 et 3dB. La deuxième section, le champ proche, est la zone allant jusqu'à 8cm de profondeur et dont la pente du gain peut varier entre 1 et 6dB. La troisième section, le champ moyen, se situe entre 8 et 20cm et a un gain fixe de 50dB. Par ailleurs, il est possible de faire varier la pente de cette zone entre -6 et 2dB. Enfin, la

dernière section, le champ loin, 20cm à 25cm, a aussi un gain fixe de 50dB mais sa pente varie plutôt de -6 à 1dB. Les simulations ont été directement exécutées sur le circuit schématique afin de caractériser le système du point de vue électrique. Les mêmes simulations ont été exécutées sous forme de tests sur la puce fabriquée à partir du modèle du dessin des masques.

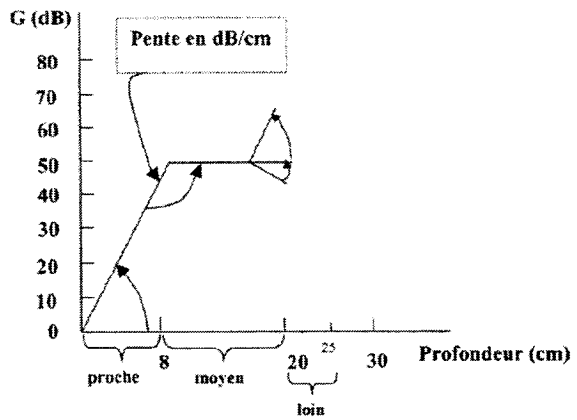


Figure 2-9 Illustration de l'échelle de compensation du TGC [13].

Le CAN convertit les signaux analogiques en données numériques. Le type de CAN choisi par notre équipe est un delta-sigma de 40 MSPS. En résumé, notre équipe présente les caractéristiques de l'intégration sur puce d'un nouveau système de réception [34]. Les résultats obtenus par l'équipe montrent le fonctionnement du TLA et du TGC en termes de performance électrique. Toutefois, le système n'a pas été modélisé et testé avec des signaux échographiques afin de valider son comportement et de vérifier son efficacité à améliorer la réception de signaux ultrasonores.

2.3.2 *Bloc numérique*

L'autre partie du système est totalement numérique. Elle est constituée du « Beamforming », du DSP et du VP. Le « Beamforming » sert à la focalisation électronique lors du balayage. Ceci a pour effet d'augmenter la résolution latérale. Il s'impose donc de présenter le balayage électronique.

La sonde utilisée en général pour ce type de balayage est linéaire et comprend plusieurs transducteurs placés un à côté de l'autre. La taille de ces transducteurs est généralement petit, ce qui indique que la zone de Fresnel est petite et que le faisceau ultrasonore tend à diverger (équations **(1.9)** et **(1.10)**). En fait, un mono élément de grande taille génère un faisceau d'ondes presque entièrement rectiligne, tandis qu'une très petite taille occasionne de la diffraction. Par contre, le comportement d'un ensemble de petits éléments agit comme un mono élément de grande taille. La Figure 2-10 montre un exemple de focalisation électronique proposé par notre équipe. Le faisceau émis (zone en dégradé de gris) va se réfléchir sur l'objet à droite de la zone couverte. Les échos provenant de l'objet vont revenir vers la sonde mais un même écho n'atteindra pas tous les transducteurs en même temps. En effet, il y aura un délai entre les transducteurs N/2, …, 2, 1 et 0. La focalisation électronique sert justement à synchroniser un même écho provenant de plusieurs transducteurs à l'aide de délai (Δ). Une fois la synchronisation effectuée, il faut additionner le signal de chaque transducteur et faire la moyenne. La Figure 2-2 présente le diagramme bloc du « beamforming ». Le bloc « Cmp » représente un comparateur qui active l'acquisition des données en provenance des CAN. Ensuite les modules représentés par « $\frac{L_n}{c/f_s}$ » indiquent des registres variables qui servent de délais pour la synchronisation des données acquises. Et le bloc « SUM » additionne tous les signaux provenant de chaque transducteur.

Le DSP est un détecteur d'enveloppe qui sert à filtrer le signal numérique obtenu du bloc « SUM ». Le schéma bloc de ce système est présenté à la Figure 2-3. Le premier module sert à la démodulation IQ, soit I le signal en phase et Q le signal en quadrature. Les blocs suivants, les « LPF » sont des filtres numériques passe bas. Ils servent à filtrer le bruit de chaque composante du signal démodulé IQ. Enfin, le dernier bloc permet la détection d'enveloppes en calculant le module d'amplitude du signal à partir des composantes I et Q. Le signal obtenu peut donc passer au traitement vidéo.

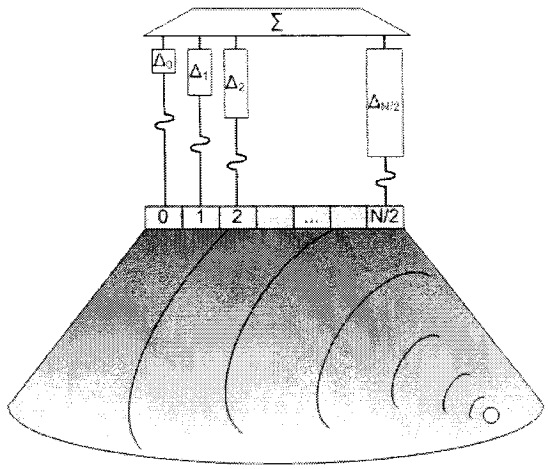
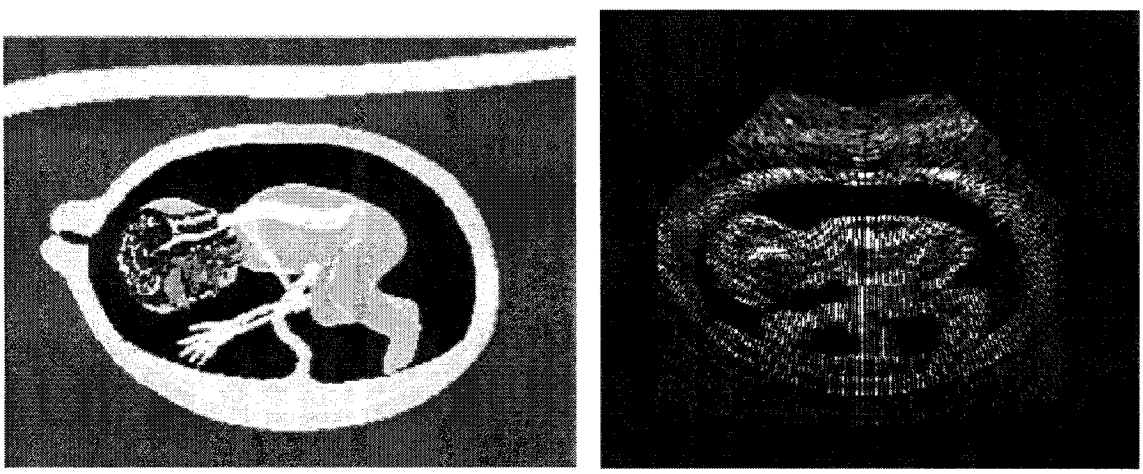


Figure 2-10 Illustration des délais variables utilisés pour le balayage électronique [17].

Le VP sert à la conversion des données polaires en données cartésiennes afin de pouvoir afficher l'image sur un LCD. Le schéma bloc est présenté à la Figure 2-5. Le bloc « Reseizing » correspond à la profondeur d'où provient le signal d'écho calculé. Le bloc « x-y plan » exécute la conversion des coordonnées polaires à cartésiennes et le bloc « Display memory » identifie la position du pixel qui sera modifié sur le LCD. L'unité de contrôle (Control unit) permet d'envoyer les données qui modifieront le pixel du LCD à la position déterminée par le « Display memory ». De plus, l'unité de contrôle gère tout le système d'interpolation de l'affichage. En fait, le côté droit de la Figure 2-6 présente le balayage électronique sans interpolation. Le problème qui résulte de cette méthode vient du fait qu'il n'est pas possible de couvrir toute la zone cartésienne à partir d'un nombre restreint de coordonnées polaires. De plus, il paraît évident que ce problème augmente avec la profondeur puisque l'écart entre deux lignes de coordonnées polaires augmente. Le côté gauche montre que l'interpolation permet de bien couvrir toute la région cartésienne. Pour ce faire, notre équipe utilise un mélange d'interpolation linéaire et de NNI.

L'architecture proposée par notre équipe se base sur huit transducteurs qui effectuent 128 lignes de balayage d'une profondeur maximale de 20cm. La fréquence de l'onde

ultrasonore utilisée est de 5MHz. Les données fournies par le CAN sont produites à un taux de 50MS/S mais seulement un échantillon sur cinq est utilisé. Ceci permet au module numérique de fonctionner à 10MHz. La Figure 2-11a présente l'image qui a servi de carte de réflectivité pour simuler la formation d'une image résultant d'un balayage mécanique sectoriel (Figure 2-11b). L'image de la Figure 2-11b est emmagasinée dans une matrice de coordonnées polaires et servira à l'évaluation des performances du système. Le fichier de test (matrice) est donc un ensemble de données numériques constitué de 128 colonnes contenants des valeurs variantes entre 0 et 40 indiquant l'intensité du pixel.



a) b)

Figure 2-11 Image utilisée pour les simulations de la partie numérique : a) image de départ et b) image en coordonnées polaires [17].

Le fait d'avoir utilisé une seule image numérisée générée par Field II pour valider l'architecture proposée n'est pas suffisant. De plus, nous constatons que les données utilisées pour faire les tests, soit la Figure 2-11b, sont déjà sous forme d'intensité de pixel. La fonctionnalité du DSP, soit la détection d'enveloppe, ne peut donc pas être testée.

Pour une profondeur de balayage de 10cm et un taux d'échantillonnage (décimation) de 10MHz nous devrions avoir 1300 échantillons pour chaque ligne de balayage. Le fichier utilisé par notre équipe n'en comptait que 640 par ligne de balayage.

Ces inconvénients nous ont donc incités à proposer un ensemble d'outils purement logiciels permettant d'être en mesure de maîtriser le fonctionnement d'un système ultrasonore avec de nombreux paramètres. En résumé, les tests qui ont été effectués ne représentent pas un bon moyen de valider le fonctionnement complet du système et d'identifier les lacunes au niveau des calculs et du codage.

2.4 Banc d'essai analogique et numérique

Maintenant que la technologie permet d'intégrer sur puce plusieurs modules de systèmes ultrasonores, il faut envisager les procédures de tests nécessaires à la validation de ces systèmes. L'un des goulots d'étranglement liés au développement de systèmes sur puce est justement l'habileté à exécuter des simulations efficaces et rapides de ces systèmes [2]. Une première technique est présentée par Loukusa à la Figure 2-12 [24]. Il propose la conception d'un banc d'essai permettant l'application d'un même stimulus à des modules de différents niveaux de développement. Il débute avec l'élaboration d'un modèle comportemental, ensuite un modèle structurel, poursuit avec un circuit schématique et termine avec le modèle basé sur une extraction du dessin des masques. Trois avantages attirent l'attention. Premièrement, il permet l'élaboration de tests avec un même stimulus pour tous. En effet, les résultats obtenus avec le modèle structurel peuvent être comparés avec ceux du modèle comportemental afin de valider le développement du modèle. Deuxièmement, il est possible d'inclure des paramètres non idéaux dans le modèle comportemental lors des simulations. Ceci permet de tester différents comportements avant même d'effectuer un design de circuit schématique. Le résultat final est un gain en temps puisqu'en général le modèle comportemental se simule beaucoup plus rapidement

que les autres modèles. Enfin, la dernière particularité est la possibilité de visualiser les résultats des tests et de pouvoir les exporter vers d'autres systèmes.

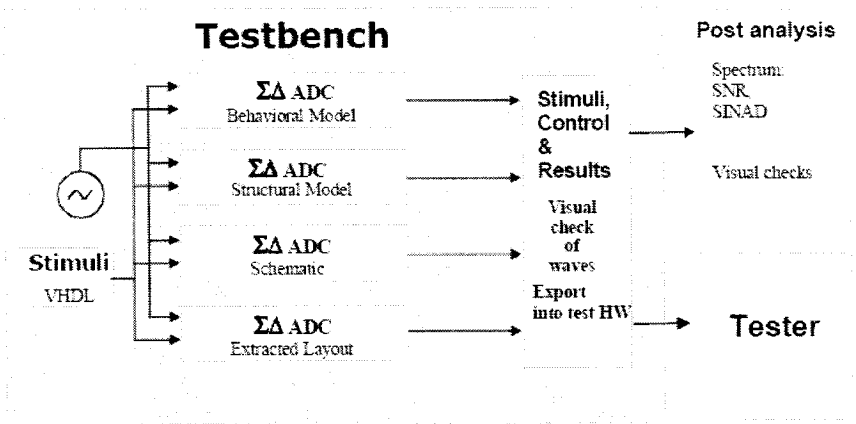


Figure 2-12 Banc d'essai constitué d'un même stimulus pour différents modèles [24].

La seconde technique est proposée par Kim [21] à la Figure 2-13. Il propose la conception d'un système de traitement ultrasonore programmable, appelé PUIP (Programmable Ultrasound Image Processor). Ce système peut être représenté comme une plateforme de tests permettant le développement rapide de nouveaux concepts de traitement de signaux ultrasonores ou l'amélioration des algorithmes déjà existants. Ce système est basé sur la technologie du multimédia. En effet, le module PUIP utilise des processeurs vidéo multimédia (MVP), ce qui permet de garder la caractéristique programmable des microprocesseurs en permettant d'augmenter la puissance de calcul. L'implémentation des algorithmes se fait à partir d'une librairie de blocks codés en langage C et en assembleur MPV afin de réduire le temps de développement et d'intégration. Le PUIP est capable de traiter différents types de données et offre le mode temps réel ou interactif (image 3D). Par contre, une fois testé, il n'y a aucune avenue facilitant l'implantation au niveau matériel (FPGA ou ASIC) des nouveaux concepts développés.

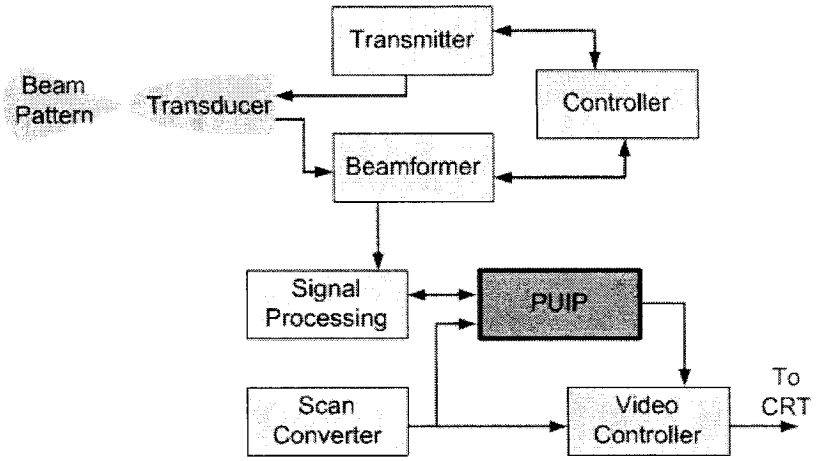


Figure 2-13 Schéma bloc de l'implantation du PUIP [21].

2.5 Outils de programmation VHDL – Verilog – Matlab

Dans le cadre du projet, nous avons choisi les langages de programmation suivants : Verilog, Verilog-A, VHDL et le script de Matlab. Le but est de pouvoir effectuer des simulations pouvant rallier des signaux analogiques et mixtes dans un même environnement, soit celui de CADENCE IC TOOLS, car notre circuit est de type mixte comme nous l'avons montré précédemment.

Le Verilog et le VHDL sont tous deux des langages de description matériel (HDL). Ils permettent de décrire un système de façon comportementale ou structurelle. Les deux langages sont très semblables, quoiqu'ils diffèrent sur la syntaxe et sur quelques autres points expliqués amplement par PECHEUX and al. [31]. Un des grands avantages de ces langages est de pouvoir implanter le code d'un système directement dans un FPGA. L'inconvénient est que ce type de langage est destiné en grande partie aux systèmes numériques.

Pour compenser cette lacune du Verilog HDL, le Verilog-A a été développé. Le Verilog-A a pour principal objectif de permettre la conception de différents types de systèmes. Sa

flexibilité se présente sous forme de « disciplines » caractérisées par la nature de ses propriétés. Ce langage est parfaitement adapté à la description comportementale et structurelle de systèmes analogiques. Par exemple, il est possible de contrôler le délai, le slew rate, faire une dérivé ou une intégrale ou de l'algèbre avec les signaux. Par contre, l'intégration sur puce des systèmes analogiques représente un défi de taille qui n'est pas résolu de nos jours [2]. Au lieu de FPGA, il faut construire des ASIC. Cette façon de faire est laborieuse. La procédure générale est présentée à la Figure 2-12 et se résume en la conception de quatre modèles : le comportemental, le structurel, le schématique et le dessin des masques. L'environnement de CADENCE IC Tools permet ce type de conceptions que l'on appelle descendantes. C'est-à-dire qu'il faut commencer avec la programmation de haut niveau (comportementale) et descendre vers le bas niveau (dessins des masques). De plus, l'environnement de CADENCE IC Tools possède les outils permettant de faire des simulations mixtes. Ce qui permet de simuler les interactions entre les systèmes analogiques et numériques.

Une partie du code Verilog et tout le VHDL ont été simulés à l'aide de Modelsim de Mentor Graphique, ceci afin d'essayer d'employer les mêmes langages de programmation pour effectuer toutes les tâches. Enfin le logiciel Matlab a surtout été employé afin de visualiser des concepts, des équations et pour faire l'affichage des images ultrasonores générées par la modélisation.

2.6 Problématiques

La problématique de ce projet de système ultrasonore vient du fait que l'interaction entre tous les modules réalisés par notre équipe n'a jamais pu être testée. De plus, lors des tests individuels de chaque module, la partie analogique n'a été testée qu'avec des signaux sinusoïdaux et la partie numérique a fait l'objet d'une absence remarquée de simulations pertinentes, ne permettant pas de prouver la fonctionnalité réelle de cette partie. Par

conséquent, il n'est pas possible de savoir si les modules sont adaptés pour un signal réel provenant d'un transducteur, si la détection d'enveloppe est fonctionnelle, si la compression vidéo est adaptée au gain du Front-End, la procédure réelle du convertisseur des signaux numériques (DSC - Digital Signal Converter) lors de l'affichage, si le vrai focus dynamique fonctionne réellement, si le système est performant et si le système peut être optimisé.

Nous proposons donc dans ce mémoire de faire la caractérisation des systèmes ultrasonores portables et de modéliser un système fonctionnant à 5MHz. L'innovation de ce projet est de concevoir un environnement de tests pour l'ensemble d'un système ultrasonore qui devrait avoir les répercussions suivantes :

- Augmenter la vitesse de vérification; permettre d'optimiser les différents modules et fournir des données qui pourront faire l'objet de comparaison, par la création d'un environnement de simulations constitué de modules de différents niveaux d'abstraction : comportemental, structurel, schématique et extraction du dessin des masques.

- Faciliter la caractérisation d'éléments piézoélectriques; évaluer les performances des « front-end » et fournir des signaux réels pour tester les modules numériques (DSP, « beamforming », « scan converter »), en développant un banc de test pour générer des images réelles qui seront utilisées dans l'environnement de simulation. Les signaux alors utilisés sont bruités et proviennent directement d'éléments piézoélectriques.

- Valider le fonctionnement d'un système ultrasonore portable de 5 MHz, en modélisant une partie du système ultrasonore développé par notre équipe de recherche et en améliorant la partie numérique du système ultrasonore proposé.

- Optimiser le système ultrasonore, en créant une méthodologie de tests qui tient compte des paramètres importants des systèmes ultrasonores, ce qui pourrait représenter un outil pour le développement de systèmes ultrasonores sur puce.

2.7 Conclusion

En somme, ce chapitre met fin à l'étude de tous les outils nécessaires à la conception de notre modèle et de notre environnement de tests. Les récents développements des systèmes ultrasonores intégrés ont été présentés, ainsi que les travaux de notre équipe de recherche Deux exemples de banc d'essai permettant d'accélérer la vérification et la conception de systèmes intégrés ainsi que les outils de simulation ont été présentés. Enfin, à la lumière de nos recherches nous avons émis des hypothèses que nous avons présentées sous la forme de problématiques à résoudre. Le prochain chapitre fait état de la conception de l'ensemble complet du système de simulation.

CHAPITRE 3

Environnement de simulation : méthodologie et conception

Aux chapitres 1 et 2 nous avons fait une évaluation de l'ensemble des développements de la technologie des systèmes ultrasonores portables afin de montrer le peu de progrès au niveau des systèmes sur puce. Il a été démontré que plusieurs éléments devraient être modifiés avant de concevoir un système fonctionnel complet sur puce. La modélisation et la caractérisation du système doivent donc être faites d'une façon méthodique afin de faire des simulations complètes, efficaces et rapides. Nous présentons dans ce chapitre l'environnement de simulation que nous avons développé pour un système ultrasonore portable. La première section fait une brève description de l'architecture de l'environnement de tests afin d'avoir une vue d'ensemble des travaux réalisés. Par la suite, nous nous concentrons sur la validation et la vérification du fonctionnement des différents modules conçus par notre équipe.

Les sections suivantes se veulent donc un approfondissement de la description des différents modules et équipements employés afin de réaliser les simulations désirées. La section 3.2 présente le banc d'essai utilisé pour recueillir les données nécessaires aux simulations. Elle comprend les fantômes, l'échographe et le système d'acquisition utilisé pour obtenir les données de simulation. La section 3.3 dévoile la conception de chaque module de tests du modèle numérique comportemental. Elle comprend la détection d'enveloppe, la conversion du balayage angulaire en données cartésiennes, l'interpolation et l'adaptation spécifique au module numérique présenté par notre équipe. La section 3.4 présente le système d'affichage des images échographiques constitué des modules de compression de données, de gestion des pixels et de reconstitution de l'image. La section 3.5 montre tous les éléments constituant le module analogique. Elle comprend les modèles comportemental, schématique et d'extraction des amplificateurs TLA et TGC et

les modules constituants le banc d'essai, soit le contrôleur et le décodeur du TGC. Enfin, la section 3.6 présente l'analyse du circuit numérique dans le but de valider son fonctionnement en programmation VHDL. Toutes les descriptions des modules sous forme de code se retrouvent dans l'annexe de ce mémoire.

3.1 Travaux réalisés

Le design du système de simulation proposé dans ce mémoire est présenté à la Figure 3-1. Il se compose de 5 sections que nous avons créées :

1.a L'acquisition de données qui utilise des équipements de laboratoire

1.b La segmentation qui est programmée en Verilog

2. Le modèle numérique comportemental qui est programmé en Verilog

3. Le module analogique qui est programmé en Verilog et Verilog-a

4. Le circuit numérique qui est programmé en VHDL

5. Le bloc d'affichage programmé en Matlab.

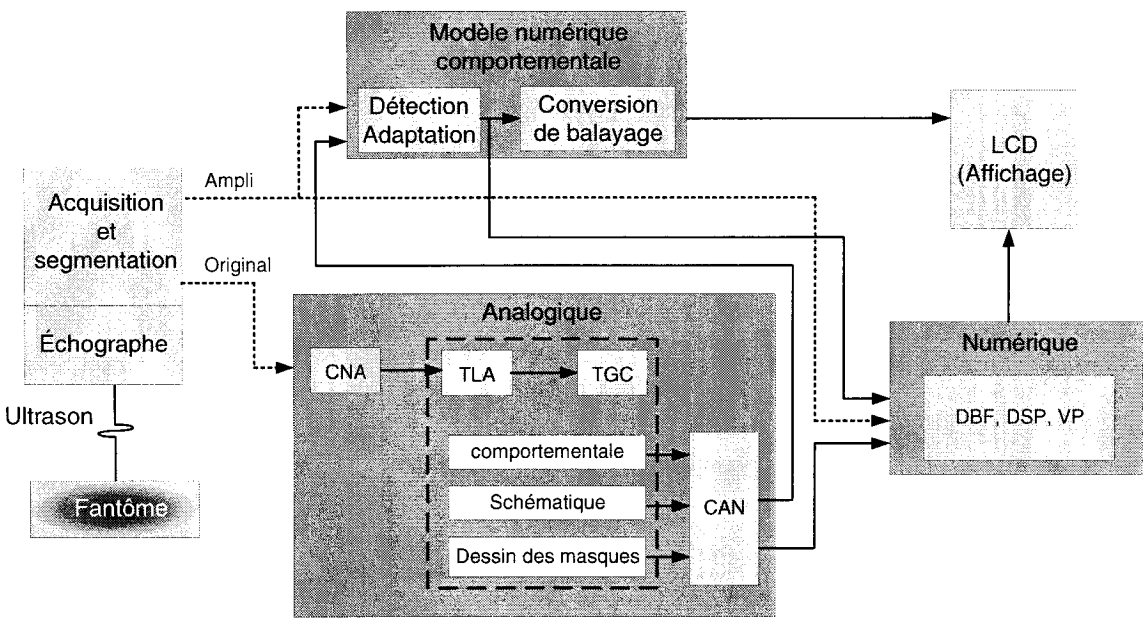


Figure 3-1 Schéma bloc de l'environnement de simulation d'un système ultrasonore.

La section d'acquisition de données permet d'obtenir un signal analogique provenant du balayage ultrasonore d'un milieu et de le convertir en un signal numérique. En premier lieu, l'onde ultrasonore est générée et interceptée par un échographe. Le milieu où se propage l'onde est appelé fantôme. En second lieu, il faut procéder à l'acquisition des données en échantillonnant le signal échographique avant et après l'étage d'amplification d'entrée, soit les signaux « Original » et « Ampli » respectivement. Ensuite les deux signaux sont acheminés au module de segmentation. Ce module sert à réduire la longueur de chaque ligne de balayage afin de garder uniquement la partie utile. Enfin les signaux peuvent être utilisés par les modules de simulations. Le signal « Ampli » est utilisé par le modèle numérique comportemental et le signal « Original » est utilisé par le module de simulation analogique. Ces signaux permettent aussi de caractériser le milieu (célérité, atténuation, longueur d'onde) et le faisceau ultrasonore (résolution axiale et latérale).

Le modèle numérique comportemental sert à afficher les données obtenues lors de l'acquisition afin de les comparer avec les résultats provenant du système ultrasonore portable (module analogique et numérique). Le modèle numérique est constitué d'un bloc, appelé adaptation, qui permet de convertir directement le signal analogique « Ampli » en données d'intensité de pixel. Comme le montre la Figure 3-1, les données obtenues en pixel peuvent alors être utilisées par le circuit numérique. Un second bloc s'occupe de la détection d'enveloppe avant d'envoyer le signal au bloc servant à la conversion des données polaires en données cartésiennes. L'image produite n'est pas interpolée mais elle permet d'avoir une bonne approximation de l'image à obtenir. De plus, la comparaison entre l'image du modèle numérique comportemental et de l'appareil de Scan View permet de vérifier la position et la variation des intensités de pixel des lignes réelles de balayage.

Le module analogique est un banc de test pour l'étage de réception du système ultrasonore présenté précédemment. Il sert à l'amplification du signal analogique provenant du transducteur. Il faut donc reconvertir le signal numérique « Original » en

signal analogique afin qu'il puisse être utilisé par les bancs d'essais du TLA et du TGC. Le signal est donc appliqué, au départ, à l'entrée des modules représentant les modèles comportementaux, schématiques et dessin des masques du TLA. Il est donc possible de comparer les signaux des différents modules entre eux. Ensuite, il faut procéder de la même manière avec le banc d'essai du TGC, en utilisant les signaux obtenus par les différents modèles du TLA. Enfin, les signaux obtenus sont alors convertis en signaux numériques par le CAN et envoyés au modèle numérique comportemental et au circuit numérique. À ce stade, il est aussi possible de comparer les signaux obtenus à la sortie du le CAN avec le signal « Ampli ». Ceci permet donc de comparer les performances de notre système avec celles de l'échographe utilisée lors de l'acquisition de données. Il en découle alors la possibilité de faire de l'optimisation.

Le module numérique contient le système de traitement de signal présenté précédemment. Les simulations peuvent être faites à partir des signaux provenant du module analogique ou du modèle numérique comportemental. Le résultat à la sortie est une image qui a subi les trois étages de traitement de signal : soit le beamforming, le DSP et le VP. Il est alors faisable de comparer les images avec le modèle comportemental. Il est important de constater que ce système utilise uniquement des données sous forme de pixels. Ce qui explique en partie la nécessité de prendre la sortie de l'ADC et de l'envoyer au modèle numérique comportemental. Ainsi, il est possible de tester le module avec deux types différents de données. Le premier type étant directement le signal analogique numérisé et l'autre provenant du modèle comportemental qui exécute un premier traitement.

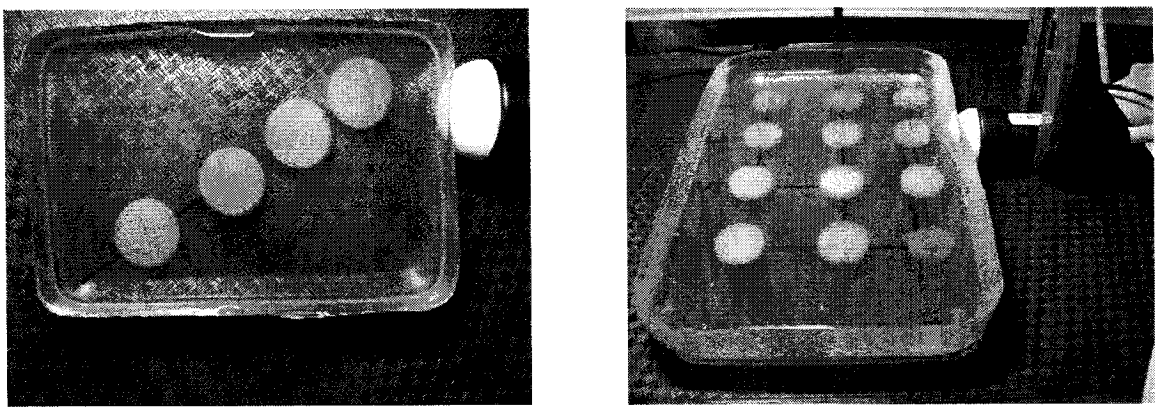
Le module d'affichage sert à la visualisation des images échographiques obtenues après la conversion des données angulaires en données cartésiennes. Il permet de faire la gestion des pixels et il peut comparer les pixels des différentes images générées. De plus, l'échographe servant à l'acquisition de données, à l'aide d'une imprimante vidéo, fournit

une version imprimée des images ultrasonores acquises. Il est donc possible, au niveau visuel, de comparer ces images avec celles générées par notre système.

3.2 Acquisition et segmentation (voir annexe, Modelsim)

3.2.1 *Fantômes*

Le fantôme est un environnement permettant de mesurer et de calculer différents paramètres liés à la propagation d'ondes ultrasonores. Il est généralement constitué d'une matrice homogène solide dans lequel il est possible de placer des inclusions. Les inclusions sont des objets dont les paramètres sont connus et qui servent à l'élaboration de motifs divers. Ce sont ces motifs qui permettent de tester les systèmes d'imagerie ultrasonore. En fait, les inclusions formant le motif sont caractérisées par une forme, un volume, une densité et une position dans la matrice. Ceci permet alors de déterminer la résolution, le gain, la distorsion et les artéfacts d'une image échographique. Tous les fantômes employés lors de ce projet sont réalisés à la main avec une matrice constituée de 8% de gélatine et d'eau. Les inclusions sont faites à partir d'agar, qui sert d'agent de diffusion. Ainsi, à l'aide d'inclusions diverses, il a été possible de générer une variété de motifs nécessaires aux différents résultats désirés. La Figure 3-2 montre les deux fantômes employés lors des simulations. Le premier est formé de quatre motifs circulaires séparés par des distances variables de 3, 6 et 12mm qui contiennent 3% d'agar chacun. Le deuxième fantôme est constitué de trois rangées de cercles positionnés à des profondeurs de 3, 7 et 12cm et dont la concentration d'agar est uniforme en profondeur mais diffère en fonction de la colonne balayée. Chaque colonne contient respectivement 1, 3 et 5% d'agar. Enfin, la dernière image présentée à la Figure 3-3 ne provient pas d'un fantôme mais du corps humain. Elle représente une section du cœur de l'auteur. Il est possible de voir l'oreillette gauche (OG), le ventricule gauche (VG) et une partie des bicuspides (BiC).



a) b)

Figure 3-2 Image des deux fantômes utilisés pour le projet : a) représente les inclusions à distance variable à 3% d'agar, b) les inclusions équidistantes à pourcentage d'agar variable.

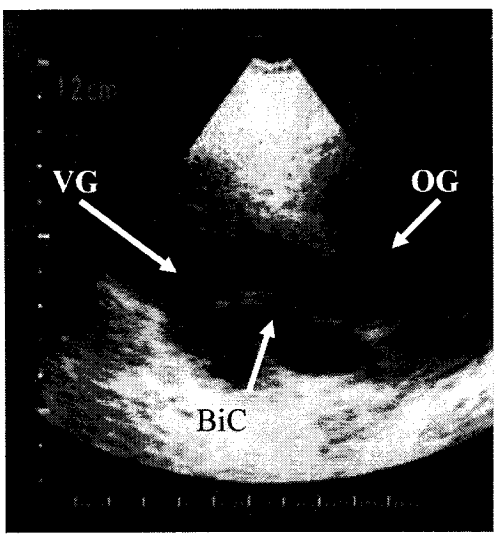


Figure 3-3 Image échographique d'une section du cœur obtenue avec le Scanview 1 : VG = Ventricule Gauche, OG = Oreillette Gauche, BiC = Bicuspides.

3.2.2 *Acquisition*

L'acquisition des données d'imagerie ultrasonore est faite à l'aide d'un échographe de la compagnie ScanView [58]. L'appareil utilisé est le Scanview1, présenté à la Figure 3-4**A**. Il possède un écran LCD, une sonde mécanique et une sortie vidéo. La profondeur maximale de balayage est de 24cm. L'imagerie est en temps réel et la sortie vidéo permet de prendre en photo l'image affichée sur l'écran LCD. Il suffit de brancher l'appareil à une imprimante vidéo.

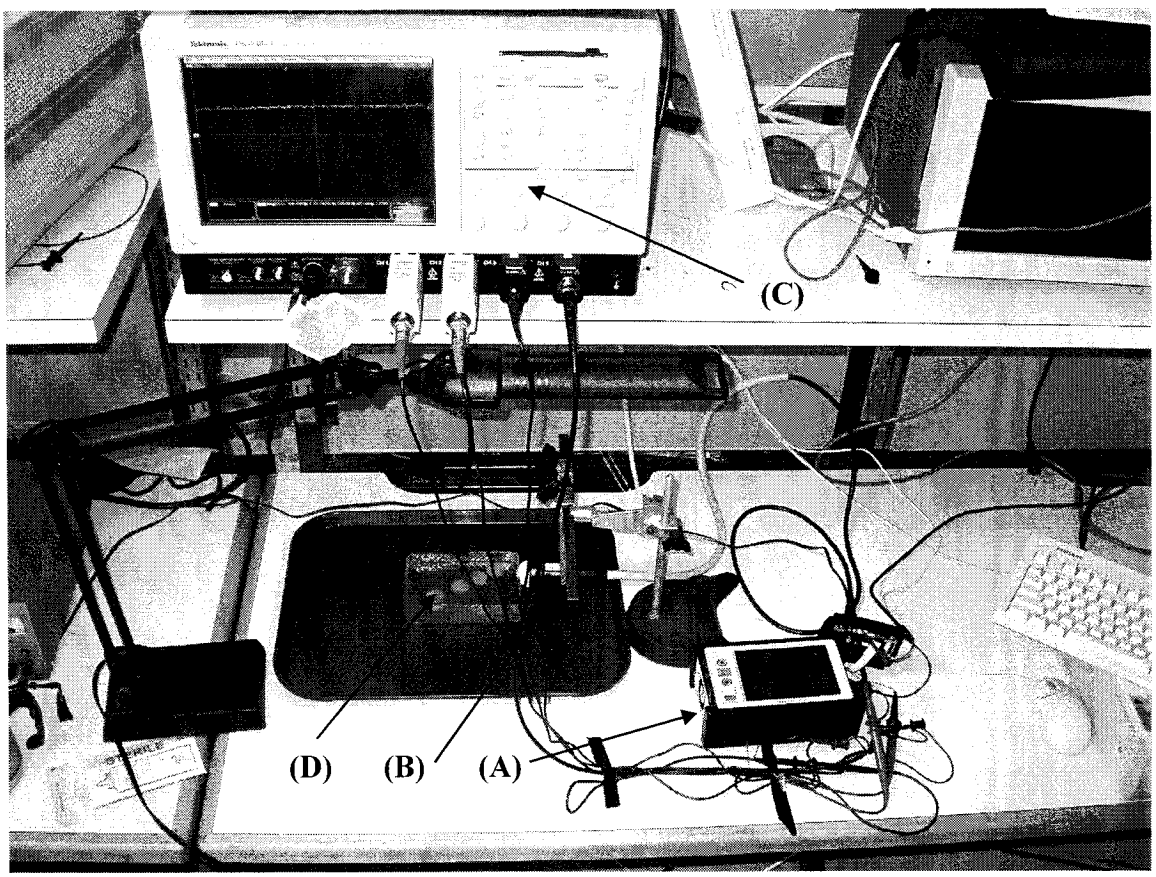


Figure 3-4 Banc d'essai de l'acquisition de données échographiques avec un fantôme : A) échographe, B) sonde, C) analyseur et D) fantôme.

La sonde, en **B**, est une GP2000 de la compagnie Interson Corp., et fonctionne de manière mécanique. Contrairement au balayage électronique, le balayage mécanique compte généralement un seul transducteur et un moteur. Grâce à ce moteur, il est possible de couvrir une zone identique à celle présentée à la Figure 2-6. Chaque pas de moteur équivaut à un certain angle de balayage. Ainsi pour un nombre x de pas de moteur, une impulsion ultrasonore est envoyée et reçue et ceci produit donc un nombre x de lignes de balayage semblables à celle de la Figure 1-5. Une contrainte importante du balayage mécanique est sa dépendance à la taille des mono éléments utilisés, qui régissent directement la longueur de la zone de Fresnel. En effet, l'objet qu'il faut sonder doit obligatoirement se trouver dans la zone de Fresnel sans quoi il y a une grande perte de précision due à la divergence du faisceau. D'après l'équation **(1.9)**, la technique du balayage mécanique ne peut pas être utilisée telle quelle par des transducteurs miniaturisés car la zone de Fresnel est proportionnelle au diamètre du transducteur. Nous allons utiliser les résultats du balayage mécanique pour tester le balayage électronique, et ce, en substituant les lignes de balayage électronique par des lignes de balayage mécanique.

Le système de balayage du Scanview1 est caractérisé par trois signaux importants : l'écho, le « fire » et le « start scan ». Deux points d'acquisition ont été établis pour faire l'échantillonnage du signal d'écho. Le premier a été placé avant l'étage d'amplification analogique de l'appareil afin d'avoir le signal « Original » provenant directement du transducteur. Le second a été placé après l'amplification analogique afin d'obtenir le signal « Ampli ». Les autres points sont le signal « start scan » qui indique le début d'une séquence de balayage et le signal « fire » qui indique le moment où l'impulsion ultrasonore est produite.

Le système utilisé pour faire l'acquisition des données est un analyseur de signal de la compagnie Tektronix, montré à la Figure 3-4 **C**. Il possède quatre canaux qui peuvent échantillonner à une vitesse de 50MHz et mémoriser plus de 80ms de données. Ceci

représente plus de 54Moctets de données enregistrées par canal, soit 4 millions d'échantillons. Sachant que le signal provenant de l'échographe est constitué de 128 lignes de balayage et que chaque ligne dure environs 375µs, il est alors possible de constater que la durée d'un balayage complet pour former une image est d'environ 48ms. Il faut donc extraire le signal désiré de la séquence d'acquisition de 80ms obtenue avec l'analyseur. Le signal analogique acquis alors, représenté par l'équation (1.12), devient le signal numérique représenté par l'équation (3.1) après l'acquisition :

$$S_{echo}(n) = A(nT_{ech})\cos(2\omega nT_{ech} + \varphi(nT)) \quad \textbf{(3.1)}$$

Où n est l'échantillon, ω la fréquence angulaire et T_{ech} la période d'échantillonnage.

3.2.3 *Segmentation*

La segmentation permet d'extraire le signal qui représente une image échographique d'un signal d'acquisition complet. Cette partie est constituée de deux modules qui permettent d'obtenir diverses données du signal échantillonné. Il est possible de choisir la précision numérique, le nombre de lignes de balayages, la vitesse d'échantillonnage par un multiple de 50MHz et le temps d'acquisition des échos qui est équivalent à la profondeur de visualisation.

Le premier module sert à l'extraction complète du signal. Les paramètres d'entrées sont les trois fichiers de signaux (« l'écho », le « fire » et le « start scan ») de 54Moctets contenant toutes les données décimales de l'acquisition. La Figure 3-5 présente les différents signaux présentés à l'entrée du module. La synchronisation s'effectue lorsqu'il y a détection du signal « start scan »; cela indique qu'il y a 17 cycles du signal « fire » avant le début des impulsions ultrasonores. Par la suite, il y a production d'une impulsion à tous les deux cycles du signal « fire »; ceci permet donc d'obtenir tout le signal échantillonné de l'écho. Par ailleurs, afin de faire des tests sur la précision de l'échantillonnage sans être obligé de refaire de l'acquisition à plus basse fréquence, nous

avons un paramètre qui permet de réduire la fréquence d'échantillonnage.

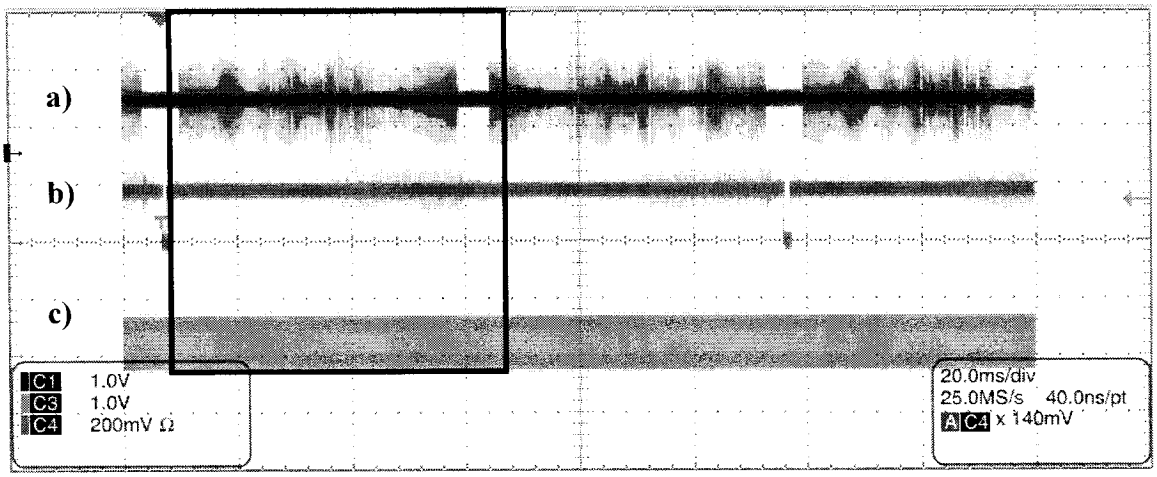


Figure 3-5 Les trois signaux de synchronisation a) écho, b) « start scan » et c) « fire ».

Le second module sert à la division du signal. En effet, chaque ligne de balayage du signal d'écho est indépendante. Il est donc pensable de traiter ces lignes en parallèle afin d'accélérer la vitesse des simulations. Il est donc envisageable de former des groupes de 128, 64, 32 ou 16 lignes de balayage équivalant à 1, 2, 4 et 8 fichiers de données. De plus, il est faisable de réduire la profondeur à visualiser en diminuant le nombre d'échantillons conservés pour chaque ligne de balayage. Voyons maintenant comment cela est possible.

La durée d'une ligne de balayage est d'environ 375µs. L'échantillonnage à 50MHz de cette ligne produit donc $375us * 50MHz = 18750$ échantillons. Ainsi, le nombre total de données pour un balayage complet est de $128lignes * 18750 = 2.4$ millions d'échantillons. Puisque les tests se font pour des profondeurs de 5, 10 et 15cm, il est inutile d'utiliser la banque complète de données. Car une ligne de 375µs équivaut à une profondeur d'environ 30cm. Pour calculer le temps relié aux profondeurs de visualisation désirées, il suffit de se référer à l'équation **(1.11)** et de l'appliquer à la définition de l'échantillonnage pour obtenir l'équation **(3.2)** :

$$Nb_Échantillon = Temps * f_s = \text{Distance} * \frac{f_s}{c} \quad \textbf{(3.2)}$$

où *Nb_Échantillon* est le nombre d'échantillons pour le *Temps* qui est la durée d'un aller retour d'un écho et f_s la fréquence d'échantillonnage. Donc 5, 10 et 15cm donnent respectivement 3250, 6500 et 9750 échantillons par ligne. Un échantillon à tous les 20ns, soit une période de f_s, équivaut à une profondeur de 30.8um. Par contre, dans le cas du signal ultrasonore, il faut tenir compte de la durée du signal qui est un aller - retour. Le signal échantillonné à 20ns représente donc en réalité un écho ayant parcouru 15.4um, soit $\frac{20ns}{2} * c$.

Le temps, qui détermine la position d'une interface, est donc commun aux deux signaux « Original » et « Ampli », ce qui n'est pas le cas de l'amplitude des signaux. En effet, le signal « Original » à une tension qui varie entre 5µV et 500mV comparativement au signal « Ampli » qui varie entre 0 et 1.8 volt. De plus, le signal « Ampli » a une tension de décalage d'environ 0.9 volt, tandis que le signal « Original » n'en possède aucune. Cette information est très importante car elle doit être appliquée lors de la conception des modules d'amplification analogique et lors de la numérisation des signaux.

3.3 Modèle numérique comportemental (voir annexe, Modelsim)

Le modèle numérique comportemental sert au traitement numérique des données provenant soit du système d'acquisition, soit du module analogique. Il est constitué de deux modules qui saisissent l'enveloppe d'un signal numérique et qui font ensuite la conversion des valeurs de ce signal, de polaires à cartésiennes. Ceci permet donc d'obtenir une image qui peut être affichée sur un écran. Le schéma bloc est présenté à la Figure 3-6.

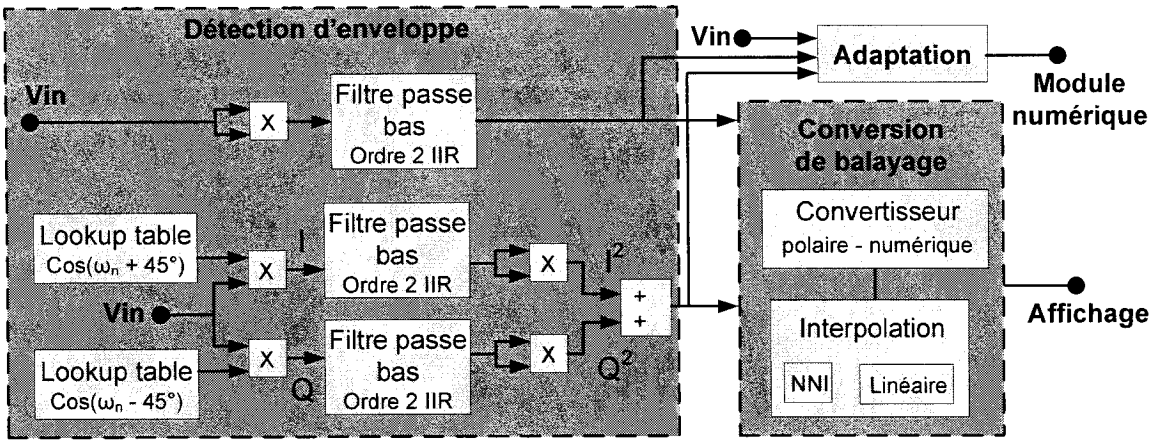


Figure 3-6 Schéma bloc des modules du modèle numérique comportemental

Le premier module sert à la détection de l'enveloppe du signal afin d'extraire la composante d'amplitude du signal ultrasonore $(A(nT_{ech}))$. Deux techniques sont présentées. La première consiste à multiplier le signal S_{echo} avec lui-même et à filtrer le signal obtenu à l'aide d'un filtre passe bas. Le filtre passe bas est un Butterworth numérique récursif (IIR – Infinite Impulse Response) d'ordre 4. La bande passante est de 3MHz avec une atténuation de 45dB à 10MHz. Cette technique est simple et permet d'éliminer une bonne partie du bruit. La multiplication sert uniquement à dissocier la composante d'amplitude $(A(nT_{ech}))$ de la porteuse à 5MHz. L'équation **(3.3)** présente le résultat de cette multiplication :

$$S_{echo}(n) * S_{echo}(n) = \frac{(A(nT_{ech}))^2}{2}[1 + \cos(2\omega nT_{ech})] \qquad \textbf{(3.3)}$$

Par la suite, il faut introduire ce signal dans le filtre passe bas afin d'éliminer la composante $cos(2\omega nT_{ech})$ et ainsi obtenir un signal égal à $(A(nT_{ech}))^2/2$. L'inconvénient de cette procédure est que le signal obtenu n'est pas exactement $(A(nT_{ech}))$. En fait, la valeur qui est obtenue à partir de $(A(nT_{ech}))^2/2$ est simplement une atténuation du signal, car l'amplitude maximale de l'enveloppe est limitée à la moitié de l'intervalle maximale de la porteuse, soit 0.9V. Cette valeur étant inférieure à 1 alors le résultat ne peut pas être supérieur à 0.9V et ainsi il n'y a aucun débordement binaire possible. Par contre, il sera souhaitable d'ajouter un gain au signal afin qu'il occupe la plage maximale de tension.

La seconde technique consiste à utiliser la modulation et la démodulation IQ. Pour ce faire, il faut décomposer le signal S_{echo} en ses composantes de phase (I) et de quadrature (Q). À l'aide de « Lookup Table » (LKT) déphasé de $cos(\omega nT_{ech} + 45)$ et de $cos(\omega nT_{ech} - 45)$, il est possible d'avoir deux signaux déphasés de 90° qui permettent l'obtention des signaux I et Q. Les équations **(3.4)** et **(3.5)** présentent la façon dont $I(n)$ et $Q(n)$ sont obtenus.

$$I(n) = S_{echo}(n) * LKT(45) = A(nT_{ech})\cos(\omega nT_{ech} + \varphi) * 2\cos(\omega nT_{ech} + 45) \qquad \textbf{(3.4)}$$
$$= A(nT_{ech})[\cos(45 + \varphi) + \cos(2\omega nT_{ech} + 45 + \varphi)]$$

$$Q(n) = S_{echo}(n) * LKT(-45) = A(nT_{ech})\cos(\omega nT_{ech} + \varphi) * 2\cos(\omega nT_{ech} - 45) \qquad \textbf{(3.5)}$$
$$= A(nT_{ech})[\cos(-45 + \varphi) + \cos(2\omega nT_{ech} - 45 + \varphi)]$$

Où $LKT(45)$ est le « lookup table » pour 45° de déphasage et φ est le déphasage entre le signal S_{echo} et le $LKT(\pm 45)$. Ensuite il faut filtrer les deux signaux avec un filtre passe bas afin d'éliminer la porteuse à la fréquence de $2\omega nT_{echo}$ pour qu'il ne reste que la variation d'amplitude avec un certain gain de $cos(\pm 45 + \varphi)$. Le filtre utilisé est le même que précédemment. Ceci devrait prendre moins d'espace que le filtre FIR (Finite Impulse Response) d'ordre 21 conçu par notre équipe. Les produits I^2 et Q^2 sont donnés par l'équation **(3.6)** et **(3.7)** respectivement. Enfin, il suffit de faire la somme de I^2 et Q^2 pour obtenir la variation d'amplitude $A(nT_{ech})^2$ (équation **(3.8)**). Par ailleurs, il suffit d'ajouter un gain afin de profiter de l'intervalle maximale des valeurs binaires comme précédemment.

$$I = [A(nT_{ech})\cos(45 + \varphi)] \Rightarrow [\ ]^2 \Rightarrow I^2 = A(nT_{ech})^2 \cos^2(45 + \varphi) \qquad \textbf{(3.6)}$$

$$Q = [A(nT_{ech})\cos(-45 + \varphi)] \Rightarrow [\ ]^2 \Rightarrow Q^2 = A(nT_{ech})^2 \cos^2(-45 + \varphi) \qquad \textbf{(3.7)}$$

$$A(nT_{ech})^2 \cos^2(45 + \varphi) + A(nT_{ech})^2 \cos^2(-45 + \varphi) = A(nT_{ech})^2 \qquad \textbf{(3.8)}$$

Le second module permet de faire la conversion des coordonnés polaires en des coordonnés cartésiennes. Il est configurable pour des profondeurs de 5cm, 10cm et 15cm, et permet de faire différentes interpolations. La conversion se fait à partir des paramètres suivants d'un balayage : le nombre de lignes, le nombre d'échantillons par ligne, l'angle séparant chaque ligne (θ), la profondeur visualisée et les dimensions du système cartésien. Par exemple, notre équipe à choisi un écran de 320 pixels de largeur sur 240 pixels de hauteur. Pour faire la conversion, il faut calculer le rapport entre la profondeur du balayage et le nombre de pixels disponibles. Le rapport F se calcule en prenant la largeur maximale du balayage et en divisant par 320 pixels. Par la suite, pour déterminer la profondeur d'un échantillon, il faut diviser la profondeur totale par le nombre total d'échantillons par ligne. Ceci donne le rapport R. Par la suite, la variation du module A de la coordonnée polaire varie en fonction de l'équation **(3.9)**. Enfin, les paramètres x et y sont obtenus en appliquant les équations **(3.10)** et **(3.11)** respectivement :

$$A = Num_Echantillon * R * F \quad \textbf{(3.9)}$$

$$x = A\cos(\theta) \quad \textbf{(3.10)}$$

$$y = A\sin(\theta) \quad \textbf{(3.11)}$$

Le rapport F donne aussi la résolution de l'image sur l'écran. Par exemple, une profondeur de balayage de 5cm, permet d'avoir une résolution d'environ 220um par pixel, soit $5cm * \cos(45^{\circ}) * 2côtés / 320 pixels$. Évidemment, l'augmentation de la profondeur fait diminuer la résolution puisque le nombre de pixels ne change pas. Pour des profondeurs de 10cm et 15cm, la résolution serait de 440um/pixel et 660um/pixel respectivement.

L'interpolation sert à combler les pixels qui sont sous-échantillonnés, comme le montre le côté droit de la Figure 2-6. Pour remédier à cette lacune, il est possible d'utiliser la technique de la NNI ou de l'interpolation linéaire. Dans les deux cas, il faut ajouter une ligne de balayage virtuel entre deux lignes de balayage réel. Par exemple, si l'angle entre

deux balayages est de 0.7 degré alors nous créons une ligne à 0.35 degré. Ce qui diffère entre les deux techniques est la valeur associée à chaque échantillon de la ligne virtuelle. La NNI signifie que les données de la ligne virtuelle créée sont des copies des données de la ligne réelle précédente. L'interpolation linéaire signifie que les données de la ligne virtuelle créée sont la moyenne des données provenant de la ligne réelle antérieure et postérieure.

L'adaptation est le dernier module de cette partie. Il permet de modifier la quantité de bits nécessaires pour le transport de l'information provenant soit du détecteur d'enveloppe, soit directement des signaux « Ampli » et du CAN. De plus, il sert d'adaptateur de signaux pour l'entrée du circuit numérique présenté par notre équipe. L'opération consiste uniquement à appliquer un facteur de gain à l'ensemble du signal. Ce facteur représente le rapport entre l'intervalle de tension du signal d'entrée et le nombre de valeurs numériques pouvant être exprimées par le nombre de bits utilisés. Par exemple, le système proposé par notre équipe utilise 40 valeurs numériques pour représenter l'intervalle de tension du signal d'entrée (0.9V), soit un facteur de 0.9/40. Par contre, notre modèle utilise les paramètres d'un CAN qui possède 10 bits de transmission. Ceci permet donc d'obtenir un facteur de 0.9/1024. La résolution est donc augmentée. Il est aussi possible de faire varier cette résolution afin de voir l'impact sur l'image engendrée.

3.4 Affichage (Voir annexe , Matlab)

L'affichage sert à la visualisation et à la comparaison des données échographiques. Il est constitué de deux modules. Le premier fait la construction de l'image en trois étapes : la compression de données, le traitement de l'affichage et l'affichage des données. Le second module sert à la comparaison des images au niveau des pixels.

La compression de données consiste à réduire la précision de l'information en fonction de l'affichage désiré. En général, l'affichage se fait avec différents tons de gris, en débutant avec le noir et en terminant avec le blanc. Puisque l'œil distingue moins de 64 tons de gris différents, ce qui correspond à des valeurs d'affichage numérique de 8 bits, il est donc possible de réduire l'information numérique de l'amplitude du signal échographique. Car après le traitement numérique, un échantillon peut facilement être représenté par 20 à 30 bits. Pour faire cette compression, deux techniques sont utilisées : le logarithme en base deux et la compression par un nombre de valeur binaire disponible. Le logarithme en base deux s'exprime par l'équation **(3.12)** :

$$C_{\log} = \log_2(S_{tr}) \text{ où } \forall C_{\log} \in N \qquad \textbf{(3.12)}$$

où C_{log} est la valeur entière compressée et S_{tr} le signal numérique à compresser. Les valeurs obtenues varient selon le nombre de bits utilisé pour représenter S_{tr}. Si l'intervalle de valeur est inférieur à 25, un facteur de deux est ajouté à C_{log} afin de maximiser l'échelle à environ 50 tons de gris. La seconde compression consiste à diviser l'intervalle des valeurs de S_{tr} par le nombre de tons de gris désirés. Par exemple, si S_{tr} est sur 20 bits alors l'intervalle de valeur est de 2^{20}, soit 1048576. Une division par 50 détermine la valeur d'un intervalle pour un ton de gris, soit $(2^{20} - 1) / 50 = 2097.5$. Il est aussi possible d'introduire facilement dans le script n'importe quelle autre technique de compression.

La gestion des pixels consiste à déterminer la valeur de l'intensité du pixel à la position x et y. Le problème vient du fait qu'il y a plusieurs échantillons pour un même pixel. Rappelons-nous que la précision d'un pixel est de 220um pour une profondeur de 5cm et que la précision de l'échantillonnage est de 15.4um. Le nombre d'échantillons par pixel pourrait donc être de 20. Ceci représente le cas critique où il est possible d'imaginer une ligne de balayage qui traverse un pixel dans le sens de sa diagonale. Sans faire le calcul, il faut aussi tenir compte du fait qu'il peut y avoir plus d'une ligne de balayage qui traverse un même pixel. Il faut donc déterminer la façon dont la moyenne des intensités de pixels est faite. Dans le meilleur des mondes, il faudrait simplement tous les additionner et diviser par le nombre d'échantillons. Par contre, ceci demande un temps

immense de calcul pour un système qui doit fonctionner en temps réel. Deux techniques sont alors proposées. La technique utilisée par notre équipe est de garder la valeur la plus élevée des intensités de pixels. La seconde technique prône plutôt l'utilisation de la moyenne de deux. Ceci consiste à additionner deux valeurs et à diviser la somme par deux pour chaque nouvel échantillon; c'est-à-dire que la valeur du nouvel échantillon est additionné à la moyenne des deux échantillons précédents et la somme est divisée par deux; ceci donne la nouvelle valeur d'intensité du pixel. Enfin, le tout est affiché à l'écran dans un cadre de dimension 320x240 pixels.

3.5 Modules analogiques du système (voir annexe, CADENCE)

Le bloc analogique sert à la partie qui traite le signal échographique avant sa conversion en un signal numérique. Il est constitué de quatre modules : le CNA, le TLA, le TGC et le CAN. Il est donc possible d'introduire un signal réel échographique bruité dans un environnement de conception analogique afin de pouvoir l'appliquer aux deux modules d'amplification.

3.5.1 *La conversion numérique à analogique CNA*

Le CNA sert à la reconstruction du signal échantillonné lors de l'acquisition. Il est présenté à la Figure 3-7 et se compose de quatre éléments : un lecteur de fichier, un convertisseur numérique-analogique, un filtre et un amplificateur opérationnel (ampliop). Le lecteur de fichier est codé en Verilog et reçoit comme paramètre d'entrée le signal « Original » numérisé. Le signal est ensuite converti en un signal analogique par le « Convertisseur » codé en Verilog-A. Enfin, le lissage du signal est effectué par un filtre passif de 25MHz de bande passante, ceci afin de garder le plus de bruit possible car le signal doit être le plus réel possible. Le dernier module est un ampliop idéal en mode suiveur programmé en Verilog-A. Il sert à isoler le circuit des modules qui seront connectés à sa sortie et permet l'ajout d'un décalage de tension. L'utilisation de 1, 2, 4 ou

8 CNA permet, grâce à la segmentation, d'augmenter la vitesse de simulation en simulant indépendamment des groupes de lignes de balayage.

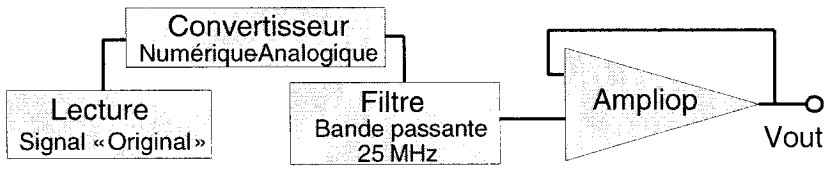


Figure 3-7 Schéma bloc de la reconstruction du signal «Original » analogique.

3.5.2 *Modèle du TLA*

Le TLA sert à accentuer de façon logarithmique la différence d'amplitudes entre les échos et le bruit. Ceci est d'autant plus important que la profondeur est grande. Le modèle du TLA est programmé en Verilog-A et est représenté par l'équation **(2.4)** développée à partir du circuit proposé par notre équipe (Figure 3-8). Les paramètres utilisés sont les suivants : A = 5.8 et V_{oL} = 0.3V. Il est aussi possible d'instaurer un délai sur le signal de sortie afin de simuler celui du vrai circuit. De plus, la simulation des

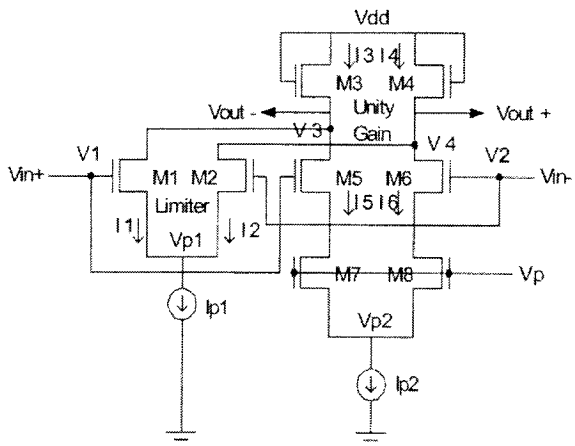


Figure 3-8 Circuit schématique d'un étage à gain dual du TLA [8].

différents modèles (comportemental, circuit schématique, dessin des masques) peut être réalisée en même temps puisqu'ils utilisent tous le même signal d'entrée. Le modèle est fonctionnel dans la technologie du CMOS 0.35µm afin de permettre la vérification du circuit réalisé par notre équipe. Il est aussi fonctionnel dans le technologie du CMOS 0.18µm.

La modélisation de cet amplificateur consiste à reproduire le résultat de sa fonction de transfert présenté à la figure 1.8, soit un amplificateur par palier de gain. Toutes les zones sont modélisées par une équation de type $y = mx + b$. Le b est déterminé par le premier degré de programmation du TGC qui peut varier entre 0 et 3dB. La pente m varie entre 1 et 6dB/cm dans le champ proche, de -6 à 2dB pour le champ moyen et de -6 à 1dB pour le champ éloigné [8]. Il faut aussi ajouter un gain uniforme de 50dB à partir du début du champ moyen. La complication initiale du TGC est le fait qu'il fonctionne en fonction du temps. L'aspect de synchronisation est donc très important puisque le TGC doit débuter avec le début d'une ligne de balayage et se terminer à la fin de cette dernière. Par la suite, il faut recommencer avec la ligne suivante. Cette partie est le rôle du contrôleur présenté à la Figure 3-9. Le module du contrôleur est codé en Verilog-A. Il en est de même pour le décodeur. Le contrôleur permet donc de faire la gestion des éléments de gain montrée précédemment avec une représentation sur 8 bits des données de contrôle. Le module de contrôle comprend uniquement un compteur et un « lookup table » qui font varier le gain en fonction de la profondeur. Le décodeur permet la conversion des données du contrôleur (8 bits) en données nécessaires à chaque étage de l'amplificateur (40 valeurs de tension). Il est constitué d'un « lookup table » de 256 lignes de 4x40 valeurs binaires reliées aux 4 étages de gain. Grâce à ces modules, tous les modèles (comportemental et circuit schématique) du TGC ont donc pu être testés. Comme dans le cas du TLA, les simulations de chaque modèle sont exécutées en parallèle.

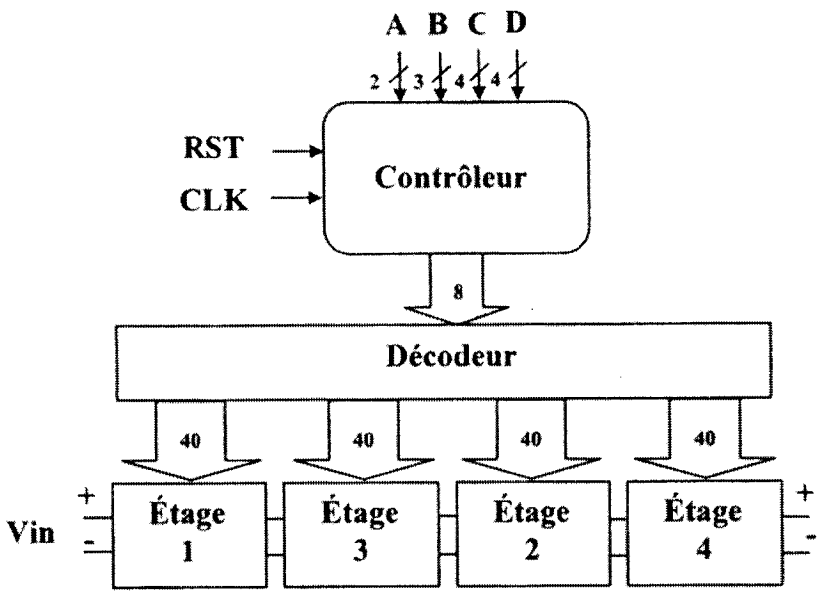


Figure 3-9 Diagramme bloc de l'amplificateur TGC à gain programmable [8].

Le dernier bloc est le CAN qui convertit le signal analogique en signal numérique. Le nombre de bits est variable, mais dans le cadre du projet, il est initialisé à 10 bits, il est programmé en Verilog-A et provient d'une banque de composants préprogrammés. Le signal obtenu est ensuite dirigé vers le modèle numérique comportemental.

3.6 Analyse du bloc numérique

Le code VHDL qui compose le bloc numérique est la version finale proposée par notre équipe. Il a été présenté au chapitre 2 afin de comprendre son utilité et son fonctionnement général. De plus, dans le cadre de ce projet, nous devions valider son fonctionnement et ses concepts de fabrication. Les concepts ont été reproduits lors de l'élaboration du modèle numérique comportemental. Leur validation est démontrée dans le dernier chapitre sur la présentation des résultats. Par contre, la fonctionnalité du système a du faire l'objet d'une analyse puisque le format des données de simulation appliquées à l'entrée (image d'un fœtus en pixel) ne concordait pas avec celles produites

par le bloc analogique (amplitude numérisée des lignes de balayage). Ainsi, au lieu de créer un module de conversion de données entre les deux blocs analogique et numérique, nous avons préféré investiguer l'architecture interne du module numérique. Ceci afin de rendre les deux blocs compatibles. Le schéma bloc du système numérique complet est présenté à la Figure 3-10.

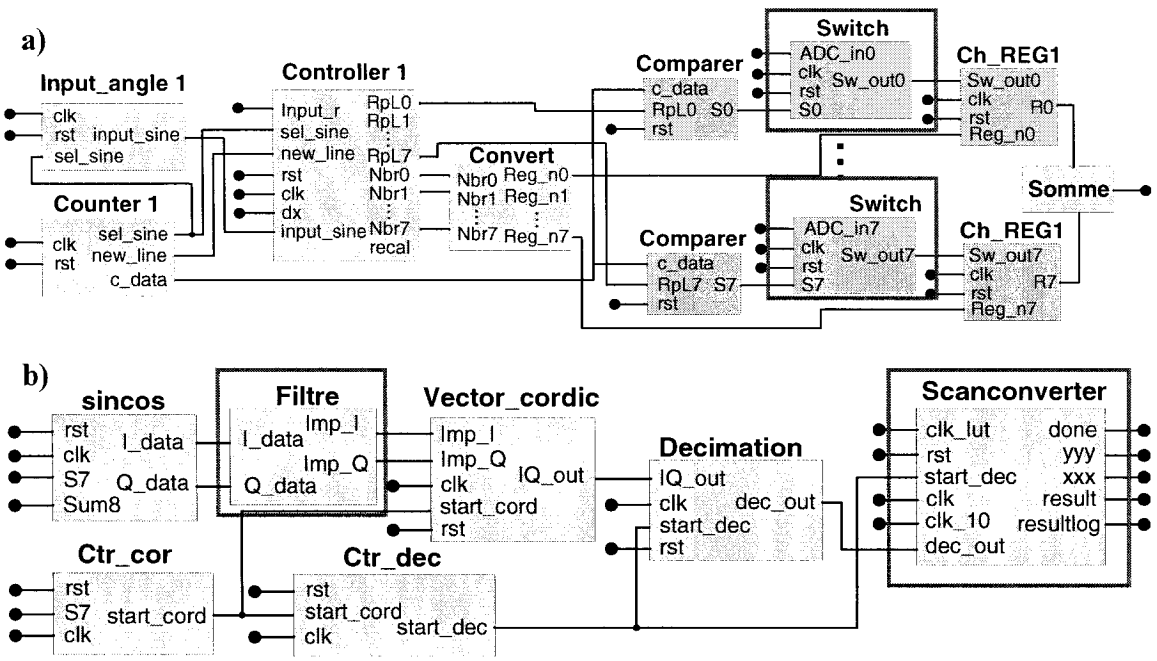


Figure 3-10 Schéma bloc du système numérique implanté en VHDL : a) le « Beamforming » et b) le DSP et le « Scanconverter ».

L'analyse est très hasardeuse car le système numérique est complexe et il faut parcourir tout le code VHDL afin de saisir le fonctionnement de chaque partie. Toujours sur la Figure 3-10, les encadrés noir représentent les endroits où nous avons remarqué des lacunes. La partie « a » représente le « beamforming » et la partie « b » la détection d'enveloppe du DSP. La première lacune ce trouve dans le module « Switch ». Au lieu de prendre un échantillon à tous les 20ns, soit à une fréquence de 50MHz, il prend un échantillon à tous les 100ns et l'utilise 5 fois. Ceci représente donc un haut taux de redondance et ne satisfait pas la précision du système. La seconde lacune se trouve dans

le module Filtre du détecteur d'enveloppe IQ. Le programme n'est pas fonctionnel. De plus, l'ordre 21 du filtre numérique employé dans le module Filtre est très élevé et demanderait beaucoup de calcul et d'espace sur un FPGA. La détection d'enveloppe n'a donc pas pu être démontrée par notre équipe.

Enfin, la dernière lacune détectable après l'analyse du programme se trouve dans le « Scanconverter ». Pour mieux saisir l'emplacement du module, la Figure 3-11 présente le schéma bloc du « Scanconverter ». La partie « a » représente la section qui génère la position des pixels et la partie « b » exécute l'interpolation. L'encadré noir montre le module d'interpolation. En effet, l'interpolation effectuée par le programme VHDL représente une NNI à droite au lieu d'un mélange de NNI

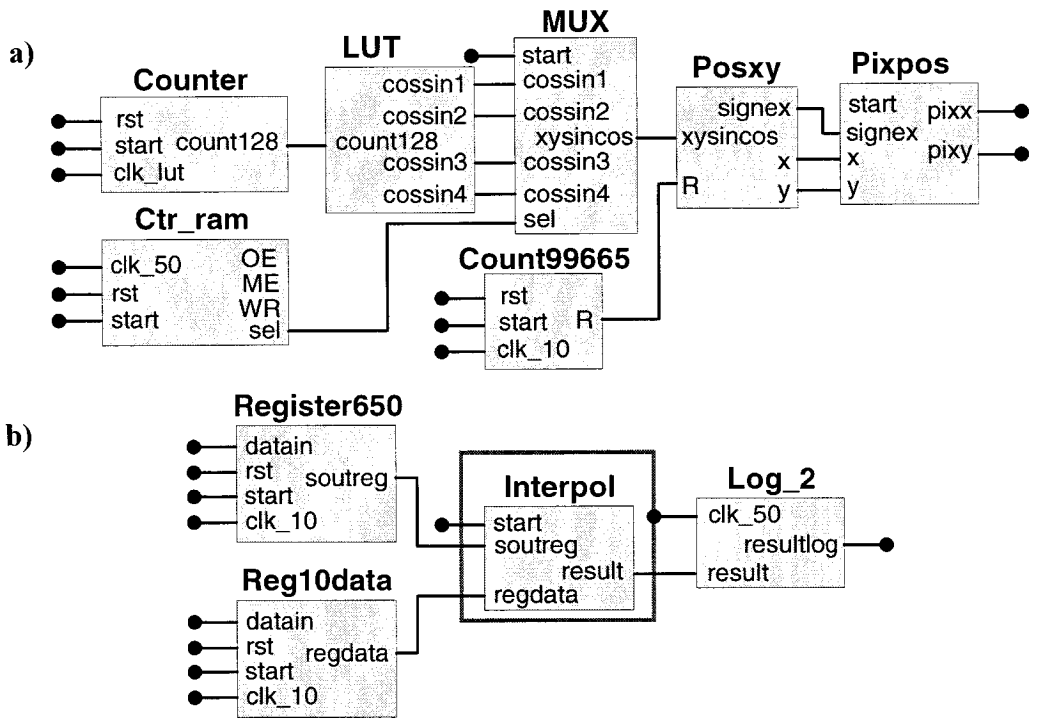


Figure 3-11 Schéma bloc du « Scanconverter » : a) calcul de la position xy et b) calcul de l'interpolation.

et d'interpolation linéaire. En se référant à la Figure 2-6, il est possible d'expliquer la NNI comme étant la réalisation de la ligne SL11, SL12 et SL13 en copiant uniquement la ligne SL1. Finalement, lorsqu'une nouvelle vraie ligne arrive comme SL2, sa valeur est

égale à la valeur la plus élevée entre cette dernière et la ligne précédente, soit SL13. Ceci pourrait expliquer le flou angulaire de la Figure 3-12, qui provient du résultat obtenu par Kassem [17].

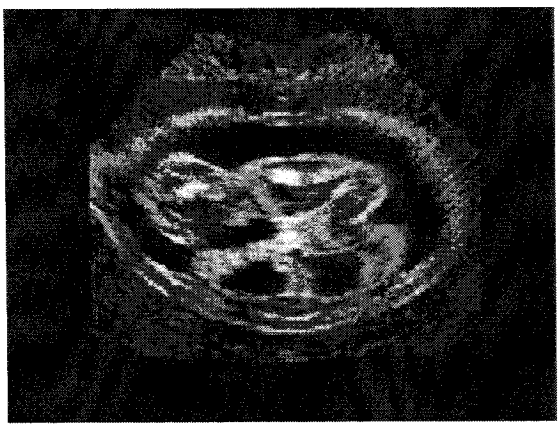


Figure 3-12 Résultat obtenu par notre équipe avec la figure 1.10a [17].

3.7 Conclusion

Ce chapitre a fait le tour de toutes les parties constituant le système d'imagerie échographique. Les différentes parties constituant le système ultrasonore à tester ont été présentées et analysées et nous avons détaillé l'aspect général de l'environnement de simulation que nous avons développé lors de ce projet. Les images paramétrées qui sont utilisées pour les simulations ont été présentées, ainsi que le processus de traitement des signaux nécessaires afin de pouvoir utiliser ces données pour valider différents modèles de modules analogiques dans l'environnement de CADENCE IC Tools. La troisième section se concentrait davantage sur le traitement numérique du signal. Développés en Verilog, les différents modules du modèle numérique rendent possible la validation de bloc spécifique programmé en différents langage : VHDL, Verilog, Verilog-A, C, SystemC, etc., ceci grâce au fait que l'information de sortie de n'importe quel module peut-être emmagasiner dans un fichier. Le bloc d'affichage rend le modèle très intéressant, car il est possible de voir les résultats d'un changement dans la chaîne

complète du système ultrasonore. Enfin, la dernière section a pu démontrer, par l'analyse du circuit numérique programmé par notre équipe, qu'il y avait des lacunes dans l'architecture et qu'elles ont pu être identifiées. Il sera donc possible d'émettre des suggestions sur les modifications à faire à partir du modèle réalisé lors du présent projet. Ceci sera présenté dans le prochain chapitre qui englobe les résultats de validation du modèle et la vérification du fonctionnement du système réel d'échographie portable.

CHAPITRE 4

Synthèse des simulations et des résultats

Ce chapitre est consacré à la description des résultats de simulation obtenus avec la modélisation du système d'imagerie ultrasonore. Le premier but est de démontrer que le modèle créé répond aux objectifs présentés lors de l'élaboration de la problématique. Le second but est de tirer des conclusions au sujet de la corroboration entre les résultats de simulation et les hypothèses de conception de notre équipe. La première section débute donc avec la présentation des résultats concernant la validation de tous les modules de la modélisation numérique qui fait le traitement du signal « Ampli » (Figure 3-1). La section deux met l'emphase sur la validation de tous les modules du modèle analogique et de tout le système (Figure 3-1). La section trois met en lumière les résultats concernant la vérification et la validation des circuits élaborés par notre équipe de recherche à l'aide de l'environnement de tests.

4.1 Résultats provenant du modèle numérique proposé

Cette section présente les résultats de chaque traitement effectué sur le signal analogique prélevé du transducteur à l'aide de l'appareil échographique de Scanview (voir le signal Ampli à la Figure 3-1). Elle débute avec l'acquisition du signal provenant du balayage de l'image de la Figure 3-2**a** en mode A et se termine avec la représentation visuelle de cette image reconstituée en mode B.

La Figure 4-1 présente l'image échographique créée par l'appareil de Scan View. Cette image correspond au balayage d'une section de la Figure 3-2**a**, où les cercles et les

espaces entre chaque cercle apparaissent clairement. Les signaux acquis lors du balayage mécanique sont présentés à la Figure 3-5 dans l'encadré noir. Le signal a de l'encadré

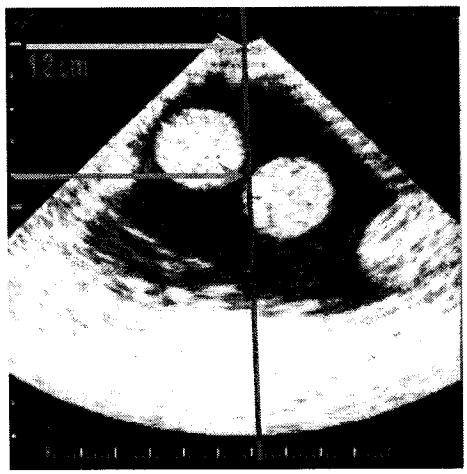


Figure 4-1 Image 1 provenant directement de l'appareil de Scanview.

noir, correspond aux signaux représentant les échos ultrasonores provenant des 128 impulsions du balayage de la sonde mécanique. Chaque impulsion est suivie d'un temps d'écoute d'environ 375µs. La Figure 4-2 présente le résultat en mode A de deux lignes de balayage pour une profondeur de 5cm après avoir effectué l'extraction, la segmentation et la reconstruction. La Figure 4-3 présente les mêmes lignes mais pour une profondeur de 10cm. Les lignes choisies correspondent aux lignes centrales du balayage, soit les numéros 65 et 66 sur l'ensemble des 128 lignes. Après la segmentation, une ligne de 5cm dure 65µs et une ligne de 10cm dure 130µs comme le montrent les figures. Les échos provenant des cercles apparaissent sur les graphiques sous la forme de grandes variations d'amplitude. La position de chaque amplitude correspond bien au tracé vertical noir de la Figure 4-1. Par exemple, l'impulsion aux environs de 10µs (Figure 4-2) possède une grande amplitude (1.36V max) et correspond à la réflexion sur la membrane de la sonde qui se trouve approximativement à moins d'un centimètre du transducteur (flèche à 0cm). L'ensemble des échos perçus autour de 60µs, donc à 4.62cm, sont reliés au contact avec le cercle. Par contre, il faut noter que l'image produite par l'appareil de Scan View élimine le premier demi-centimètre de balayage lors de l'affichage. Donc, la position de l'écho est réellement de 4.62cm par rapport au transducteur, mais, par rapport à la photo et à la membrane, il faut soustraire un demi-centimètre; cela indique donc que la

profondeur sur l'image est de plus ou moins 4cm (flèche à 4cm). En ce qui concerne le signal en mode A, le taux de bruit se situe autour de 10mV et la tension de décalage est d'environ 950mV. De plus, le système d'amplification de l'appareil amplifie le signal des échos reçu dans l'intervalle de tension de 1.8V.

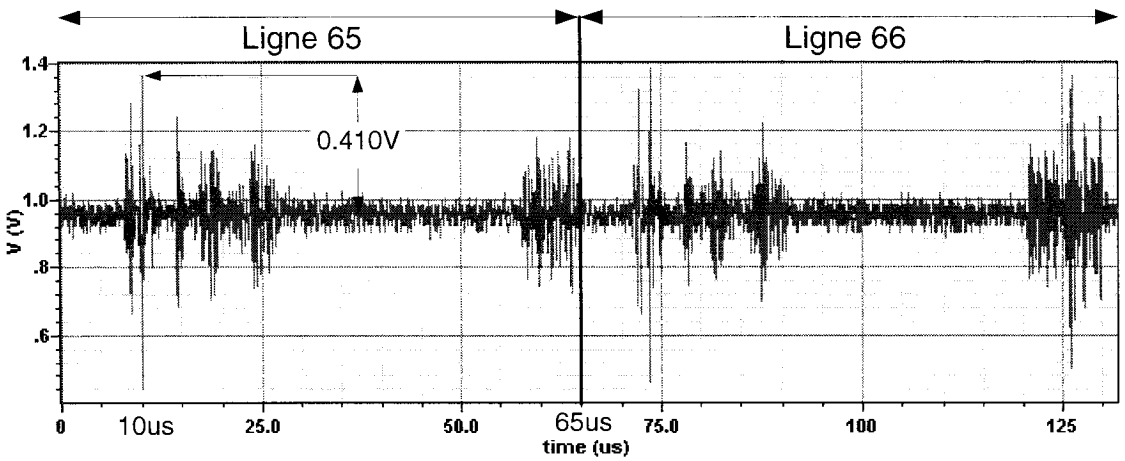


Figure 4-2 Graphique temporel des échos provenant des lignes de balayage 65 et 66 pour une profondeur de 5cm.

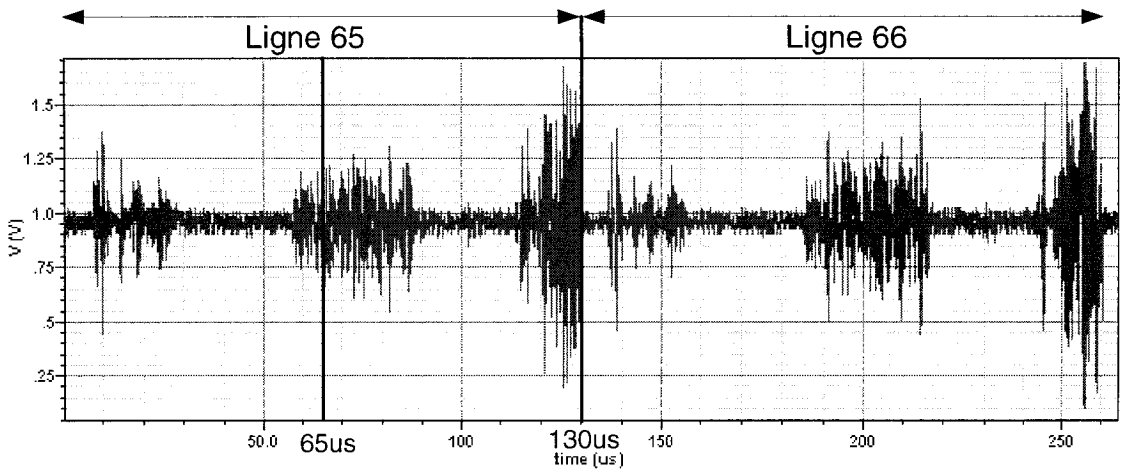


Figure 4-3 Graphique temporel des échos provenant des lignes de balayage 65 et 66 pour une profondeur de 10cm.

Le signal est ensuite traité par le détecteur d'enveloppe. La première ligne de balayage de 5cm présentée à la Figure 4-2 (ligne 65) est utilisée pour le reste des simulations. Le

premier détecteur d'enveloppe utilisé est la modulation IQ qui se retrouve dans l'architecture du système réalisé par notre équipe. Le second détecteur d'enveloppe utilisé est le détecteur simple. Les résultats des deux détections sont présentés à la Figure 4-4 et à la Figure 4-5. La constatation de départ est que les résultats des deux détections représentent bien le signal de la Figure 4-2. En raison de la forme semblable des spectres d'amplitude des deux signaux, nous pouvons affirmer que les deux enveloppes sont pareilles.

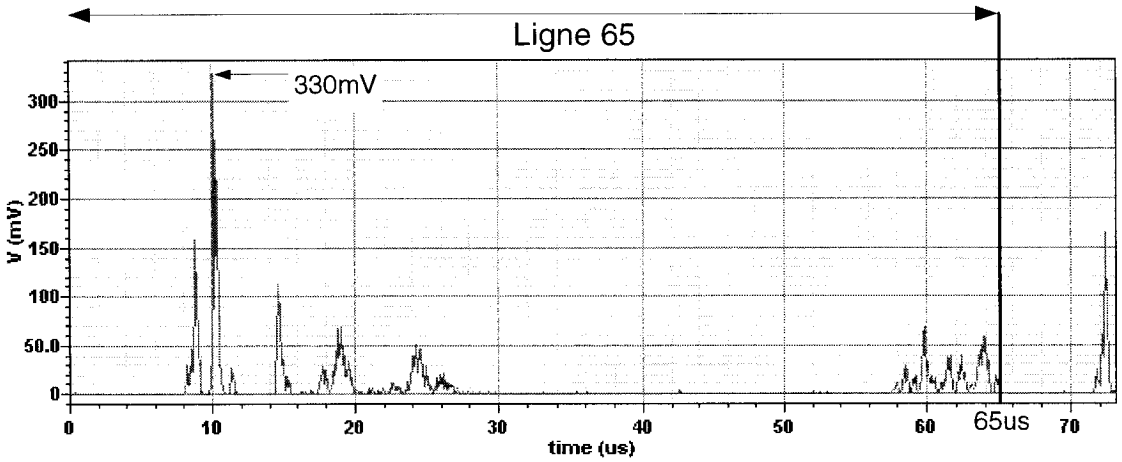


Figure 4-4 Graphique de l'enveloppe IQ du signal de la ligne de balayage 65 pour 5cm.

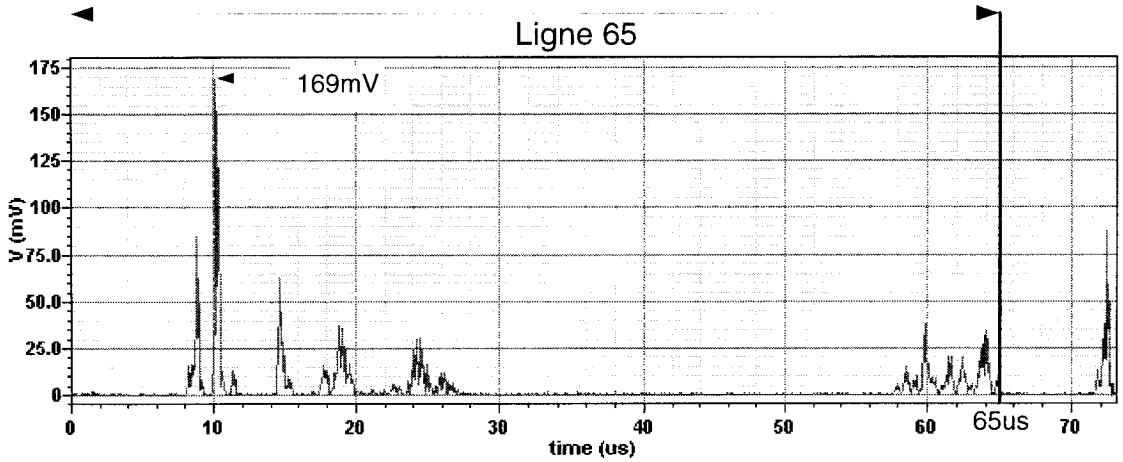


Figure 4-5 Graphique de l'enveloppe simple du signal de la ligne de balayage 65 pour 5cm.

La comparaison des tensions entre la Figure 4-2 et la Figure 4-4 montre qu'il y a un gain non unitaire par rapport aux données théoriques obtenues à l'aide de l'équation **(3.8)**. En effet, la tension maximale mesurée autour de 10µs dans les deux graphiques donne des potentiels de 410mV et 330mV respectivement. Ainsi, le carré de 410mV = 168mV au lieu de 330mV, ce qui donne un gain de 50%. Le gain du filtre conçu est donc non unitaire. Par ailleurs, les deux techniques donnent des résultats qui diffèrent beaucoup. La comparaison des résultats de la Figure 4-4 et de la Figure 4-5 montre qu'il existe un facteur d'environ deux entre les tensions. Par exemple, à 10µs, les tensions sont de 330mV et de 169mV respectivement. Ceci s'explique par les résultats des équations **(3.3)** et **(3.8)** qui diffèrent d'un facteur de deux. De plus, le signal du détecteur simple montre une légère composante de bruit tandis que la détection IQ en est exemptée. La Figure 4-6 et Figure 4-7 montrent un agrandissement en amplitude et en temps des signaux des techniques simple et IQ afin de mieux voir la différence entre les deux signaux.

La différence majeure entre les deux signaux concerne la présence de la composante fréquentielle de la porteuse de 5MHz. En fait, cette composante fréquentielle est d'environ 3.3MHz, car la sonde de Scan View produit un signal autour de 3.5MHz. Le filtre n'est donc pas adapté à cette fréquence, ce qui laisse paraître la porteuse. Par ailleurs, la comparaison des deux signaux montre que la technique IQ permet une meilleure élimination de la porteuse que la technique simple. Une autre différence importante est le rapport signal sur bruit. Le grand désavantage de la technique simple est de réduire ce facteur par un coefficient de 2. Ceci est une donnée non négligeable vis-à-vis un système qui demande une bonne précision. En fait, cette partie vient déterminer directement la résolution de l'image. Si la variation du signal occasionnée par la porteuse est trop grande, cela influencera directement l'intensité du pixel qui y est associé et cela aura pour résultat une image floue. Évidemment, au niveau système, il faut choisir entre une technique simple et facile d'implantation et une technique plus complexe mais avec un meilleur rendement.

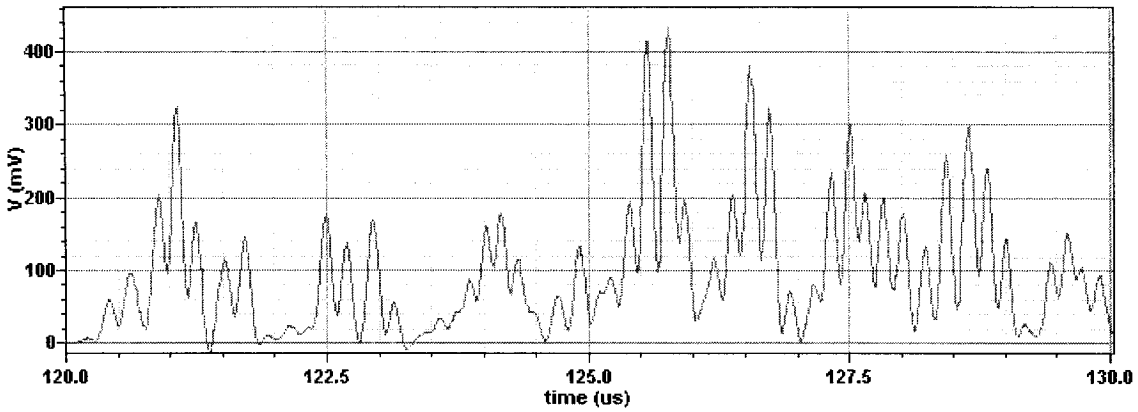


Figure 4-6 Signal de l'enveloppe simple agrandi.

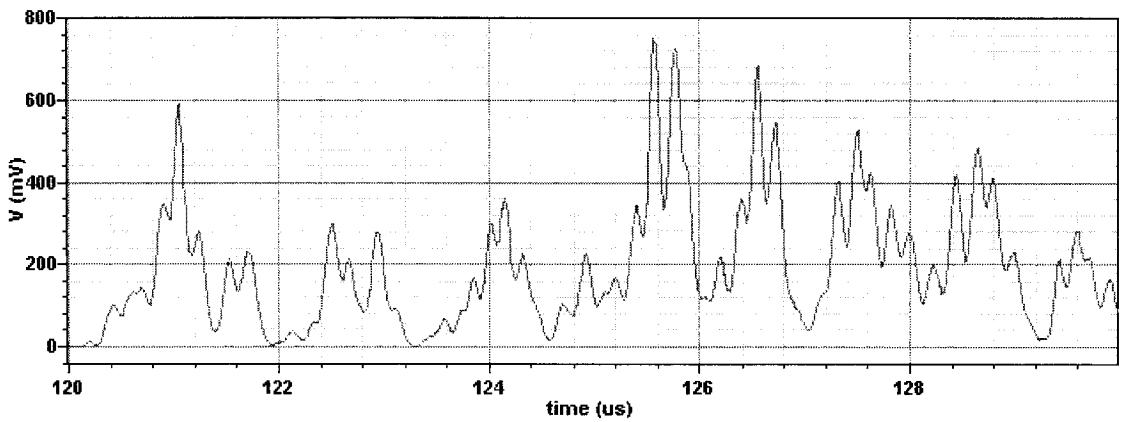


Figure 4-7 Signal de l'enveloppe IQ agrandi.

4.2 Résultats du module d'affichage

L'affichage comporte la partie de conversion des données angulaires en données cartésiennes, l'interpolation et l'affichage de l'image obtenue. Les simulations sont appliquées à deux images différentes. L'une provient directement du traitement numérique précédent (Figure 4-1) et l'autre est une image numérisée d'un fœtus (Figure 2-11b). Cette dernière provient d'un logiciel, qui permet de convertir une image quelconque en vecteur de coordonnées angulaires numérisées équivalent à un balayage mécanique.

La Figure 4-8 présente une image homogène sans interpolation (Figure 4-8a) et trois images obtenues avec les autres techniques d'interpolation. L'image en a est constituée uniquement des 128 vecteurs de balayage convertis en coordonnées cartésiennes sans interpolation. Les images b et c sont obtenues en effectuant respectivement une interpolation linéaire et une NNI. Les deux techniques emploient 256 vecteurs angulaires, soit 128 réels et 128 provenant de l'interpolation. Par contre, ce qui les différencie concerne le calcul des lignes interpolées. Une lacune de ces techniques est que la zone de balayage n'est pas entièrement couverte, ce qui indique une mauvaise définition latérale de l'image en fonction de la profondeur. Enfin, l'image d montre que le mélange de la NNI et de

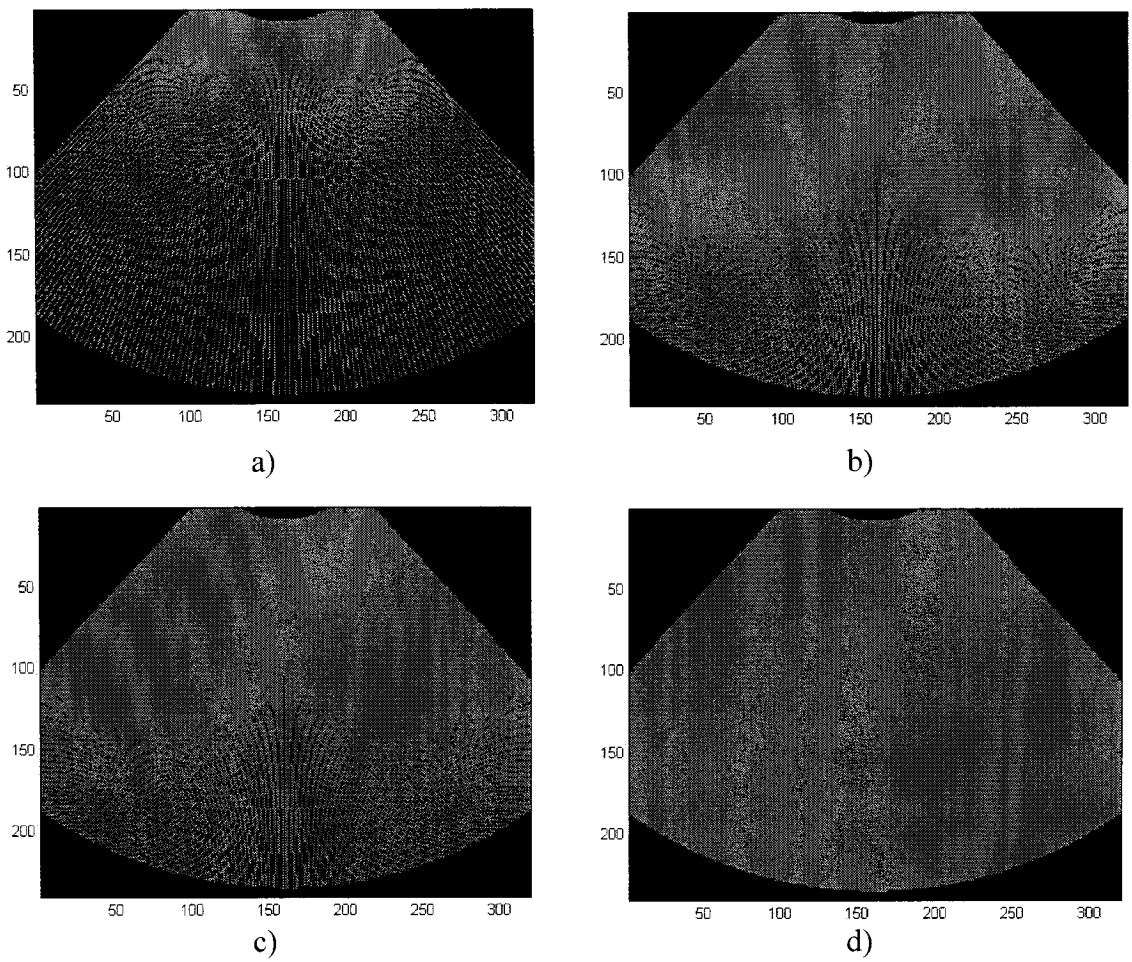


Figure 4-8 Image homogène : a) sans interpolation, b) interpolation linéaire, c) NNI et d) interpolation mixte.

l'interpolation linéaire permet de couvrir toute l'image en utilisant 512 lignes de balayage, soit 128 lignes réelles, 128 en interpolation linéaires et 256 lignes en NNI. Ceci permet de conclure que le nombre de lignes nécessaires à l'obtention d'une bonne image est de 512 pour une profondeur de 10cm. Par contre, pour une faible profondeur, il survient un problème de surreprésentation d'intensité de pixel qui est en partie résolu par la compression de pixel.

La Figure 4-9 montre le résultat de l'image 1 (Figure 4-1) acquis et traité par le modèle comportemental proposé. À première vue, elle ressemble bien à l'image 1 mais avec un degré de déformation assez élevé. Ceci est dû à deux grands facteurs. Le premier est lié au pas angulaire du balayage mécanique de l'appareil de Scan View qui est différent du pas utilisé pour notre modèle. Le second facteur provient du fait que l'image obtenue par l'appareil de Scan View est de 60 degrés et non pas de 90, ce qui explique que l'image penche légèrement vers la gauche. Les simulations ont donc été compensées pour le

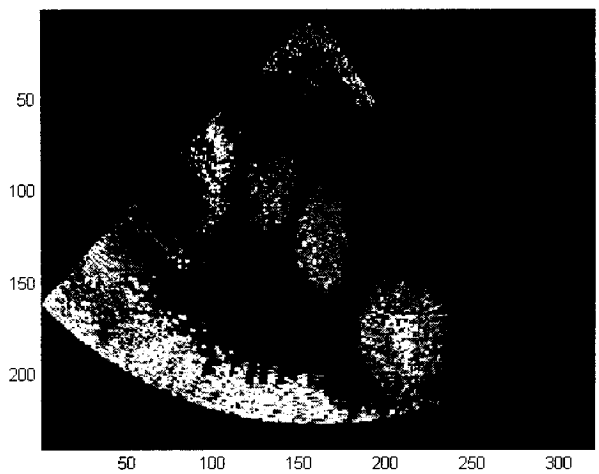


Figure 4-9 Résultat image 1 réel, obtenu avec le modèle et sans modification.

premier facteur mais pas pour le second, car le modèle est conçu pour des balayages de 90 degrés. Les paramètres de l'image sont : un filtrage IQ, une compression linéaire des données, une interpolation mixte et une méthode de compression où le pixel le plus intense est gardé.

La Figure 4-10 présente l'image 1 avec le pas angulaire modifié. Les images ont été filtrées par la modulation IQ avec une compression linéaire des données et une méthode de compression où le pixel le plus intense est gardé. Il y a uniquement l'interpolation qui varie. L'image en a est la référence car elle n'a pas été interpolée. Les images en b et en c correspondent à l'interpolation linéaire et à la NNI respectivement. Enfin, l'image en b est le résultat de l'interpolation mixte et elle montre une image plus uniforme que les

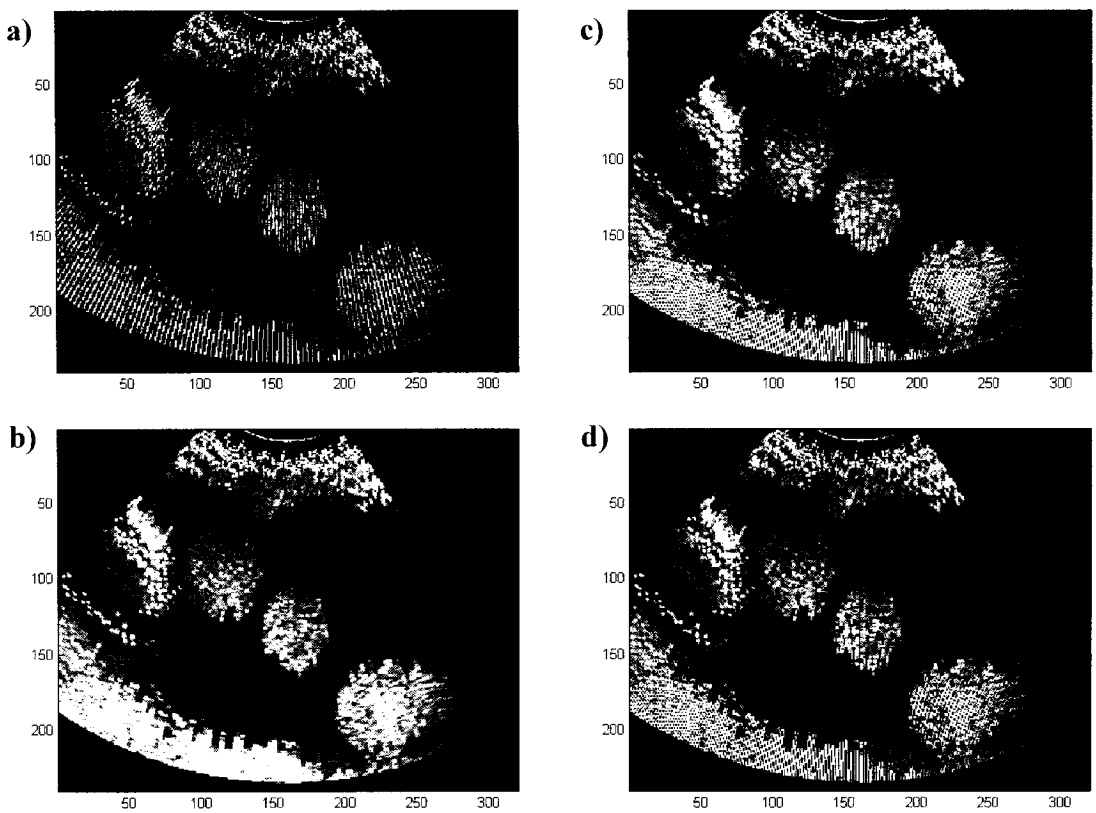


Figure 4-10 Image 1 : a) sans interpolation, b) avec interpolation mixte, c) avec interpolation linéaire et d) avec NNI.

autres. Les illustrations portent à croire qu'il n'y a pas de différence entre l'interpolation linéaire et la NNI (c et d). Pourtant, il y en a une qui est très subtile : l'interpolation linéaire amène une fluidité dans l'image en estompant graduellement les contours entre deux zones d'intensités différentes. Ce qui n'est pas le cas de la NNI.

La Figure 4-11 présente la même image qu'à la Figure 4-10d mais avec l'utilisation de la détection d'enveloppe simple. La différence se situe effectivement dans l'intensité générale de l'image qui est plus dégradés et qui montre l'apparition sporadique de pixels de grande intensité.

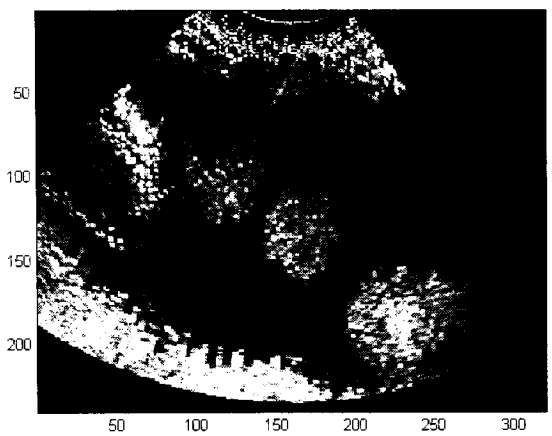


Figure 4-11 Image 1 avec un filtre simple et une interpolation mixte.

La Figure 4-12 présente aussi la même image que la Figure 4-10b mais avec l'utilisation de la compression logarithmique en base 2 au lieu de la compression linéaire. La figure montre une image floue et très sombre. Ceci s'explique par le fait que l'intervalle des valeurs d'amplitude liée à l'intensité d'un pixel augmente de façon logarithmique, donc sur une grande plage de valeurs d'amplitude pour un même ton de gris.

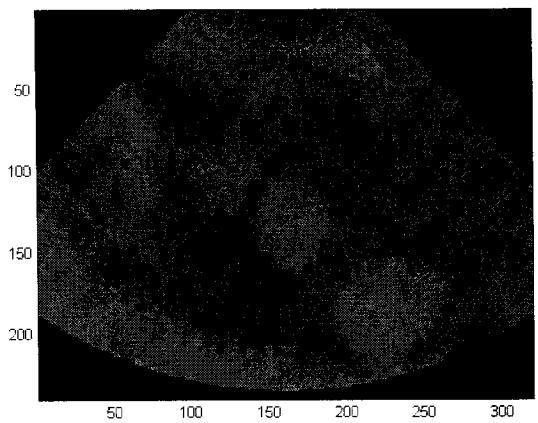


Figure 4-12 Image 1 obtenue après une compression en log à base 2.

La prochaine partie utilise l'image numérisée afin de tester la compression des données des pixels en appliquant différentes méthodes : garder le dernier pixel, la moyen de 2 ou garder le pixel le plus intense. La Figure 4-13a présente l'image du fœtus en 128 lignes afin de montrer que la reproduction est identique à celle produite par notre équipe (Figure 2-11b). La particularité des prochaines images est qu'elles utilisent toutes l'interpolation mixte, ce qui n'est pas le cas pour les résultats présentés par notre équipe, qui sont basés sur la NNI seulement.

La Figure 4-13b présente la méthode de compression qui garde le pixel le plus intense. L'image du fœtus est bien reproduite, mais elle n'est pas très claire et surtout elle est très intense en termes de luminosité. La Figure 4-13c présente la compression par la moyenne de 2. L'image du fœtus est bien mais elle reste floue. La Figure 4-13d présente la méthode de compression qui garde le dernier pixel. L'image du fœtus est encore bien affichée mais elle semble un peu granuleuse. La comparaison de cette dernière image avec celle de la Figure 3-12 montre que le modèle reproduit parfaitement le fonctionnement du module numérique de notre équipe et de l'ensemble du traitement numérique.

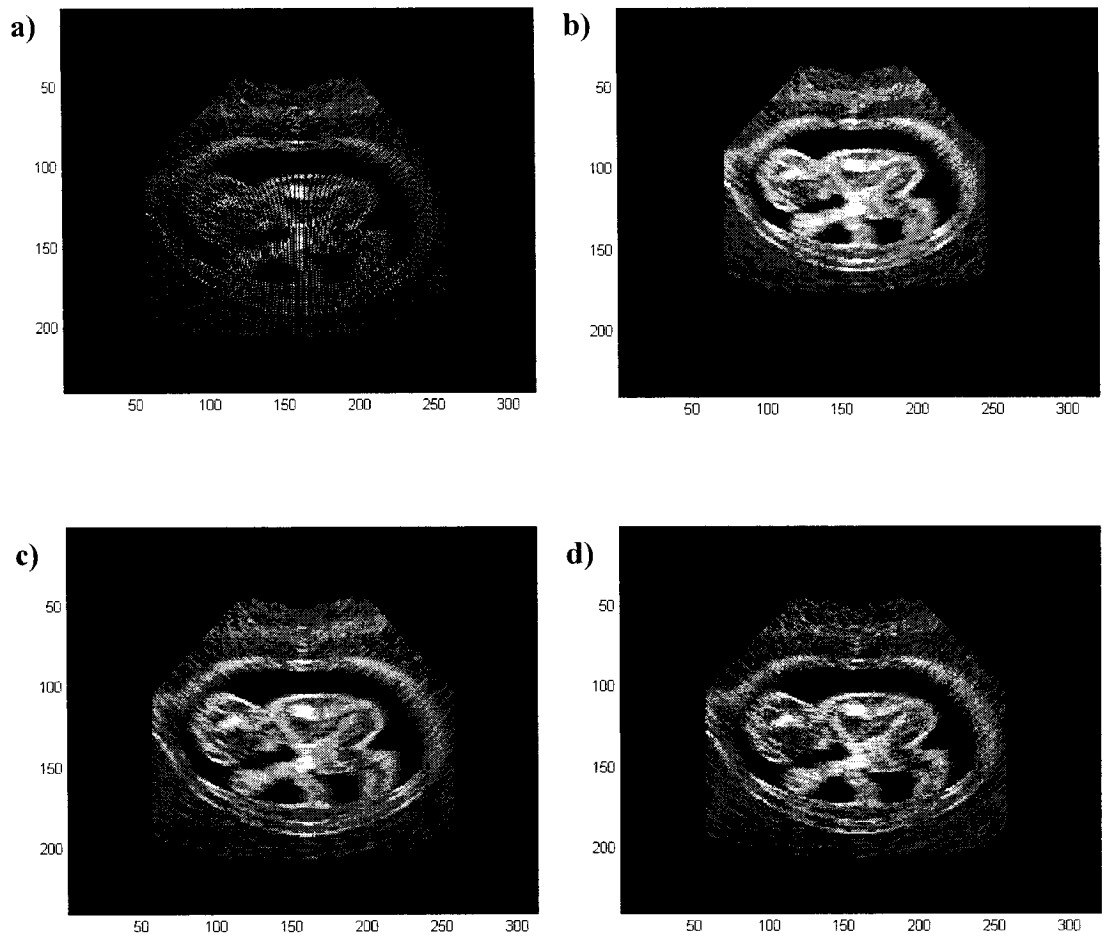


Figure 4-13 Image numérisée du fœtus produite par notre modèle : a) sans interpolation, b) sélection du pixel le plus intense, c) la moyenne de 2 et d) le dernier pixel est gardé.

4.3 Traitement analogique et numérique du signal

Cette section présente les résultats de simulations obtenus lors du traitement analogique du signal prélevé du transducteur à l'aide de l'appareil échographique de Scan View (voir le signal Original à la Figure 3-1). Elle concerne les modules d'amplification TLA, TGC

et ADC. De plus, cette section valide le traitement complet du signal : de son acquisition jusqu'à son affichage en mode B.

Avant d'effectuer des tests sur le signal directement acquis, il faut voir si l'information se trouve bien dans le signal et qu'il n'est pas trop bruité. La Figure 4-14 montre le signal échantillonné « Original » utilisé pour les prochaines simulations. Le signal est en fait le même que celui de la section précédente (ligne 65 de la Figure 4-3) mais sans amplification analogique. Le signal est un écho qui provient directement du transducteur et dure 130ms,. ce qui représente une profondeur de 10cm. Ce signal a aussi été filtré par un passe bande du 2^{e} ordre afin d'éliminer du bruit de basse fréquence indésirable et afin d'améliorer le signal. En effet, l'un des problèmes de l'acquisition de données, à même le transducteur, est l'apport important en bruit que cela amène. En fait, le bruit limite la détection des échos de moins de 5mV. Ceci cache donc beaucoup d'information indispensable à la formation de l'image.

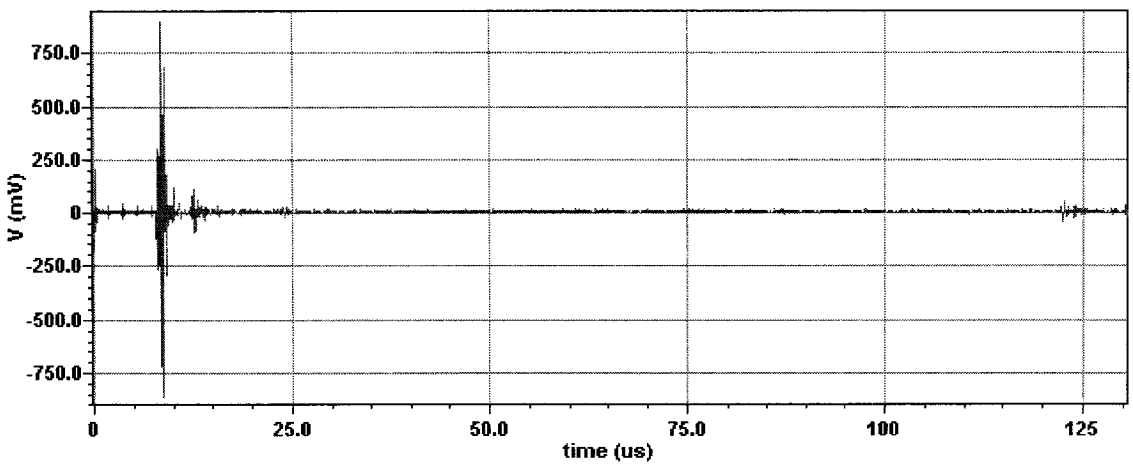


Figure 4-14 Signal « Original » après sa reconstitution et son filtrage pour une profondeur de 10cm.

4.3.1 *Simulation du modèle du TLA*

La Figure 4-15 montre la fonction de transfert du modèle comportemental du TLA conçu à partir des équations mathématiques de notre équipe. La comparaison avec la fonction de transfert de la Figure 2-8 montre une disparité dans la pente du signal. En fait, le modèle indique que l'amplification du signal est beaucoup plus grande que prévu. La Figure 4-16 montre le résultat du traitement du signal échographique par le modèle comportemental du TLA. La figure permet de constater que le signal est complètement saturé en gain. Il est alors possible de s'attendre à ce que la simulation du circuit schématique donne le même résultat. Par contre, si le gain du TLA est largement diminué et que les paramètres sont légèrement modifiés, il est possible d'obtenir un résultat significatif et intéressant. La Figure 4-17 montre ce résultat.

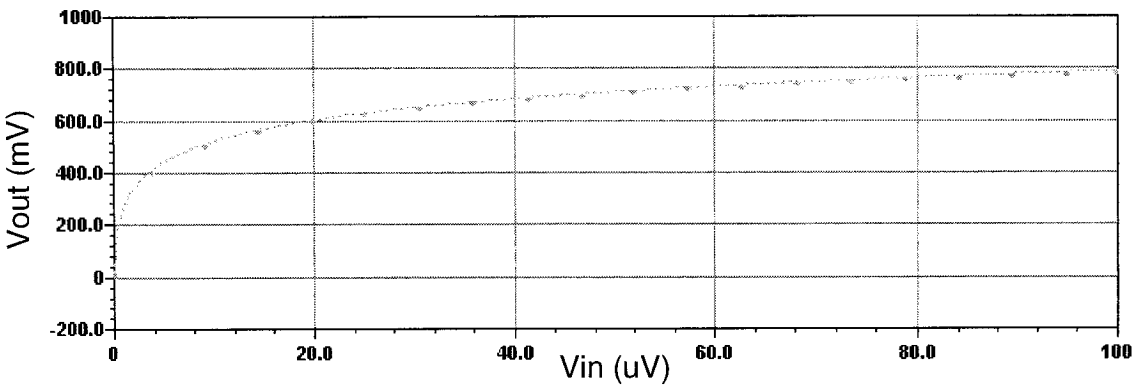


Figure 4-15 Fonction de transfert du modèle comportemental du TLA.

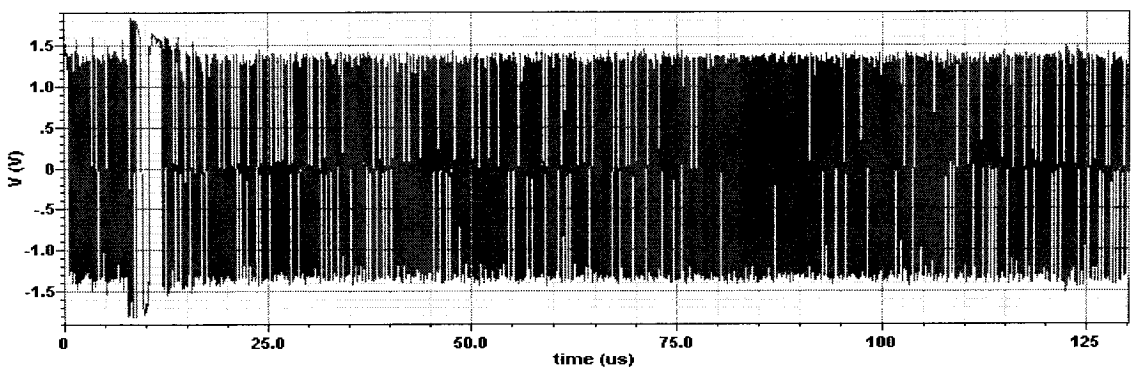


Figure 4-16 Signal « Original » après l'amplification du TLA.

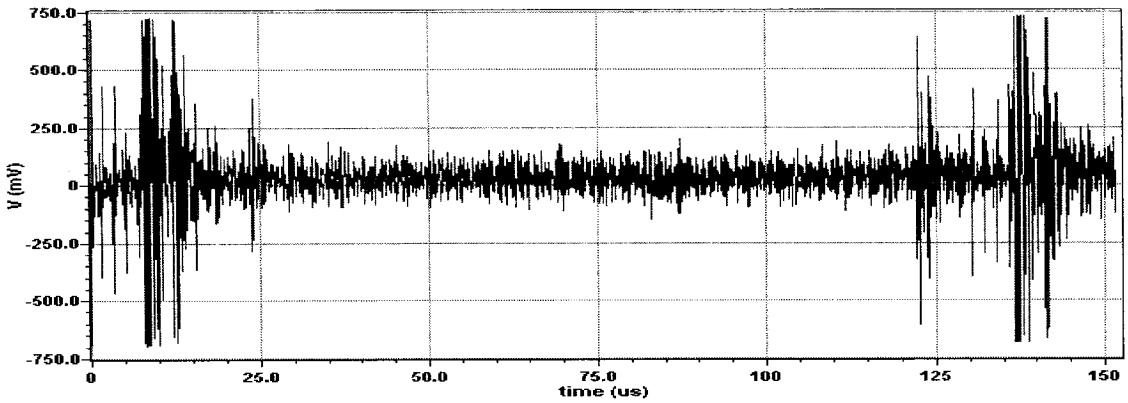


Figure 4-17 Résultat du TLA lorsque son gain est diminué.

4.3.2 *Simulation du modèle du TGC*

La Figure 4-18 présente la fonction de transfert du TGC pour le cas particulier de sa programmation en gain maximal. Contrairement au résultat présenté par notre équipe (Figure 2-9), la fonction de transfert du TGC en fonction du temps est caractérisée par des paliers de gain. Ceci aura un impact non négligeable sur le reste des simulations. Puisque le TGC est programmable, il existe alors plusieurs combinaisons possibles de paramètres. La validation du modèle nécessite seulement deux combinaisons : la valeur maximale et la valeur moyenne. La valeur minimale n'a pas été prise en compte car elle n'apporte aucune amplification. Les valeurs des combinaisons de gains sont présentées au Tableau 4-1.

Tableau 4-1 Gain programmer des champs du TGC.

Champs	Valeur maximale (dB)	Valeur moyenne (dB)
Initial	3	1
Proche	6	2
Moyen	2	-6
Éloigné	1	1

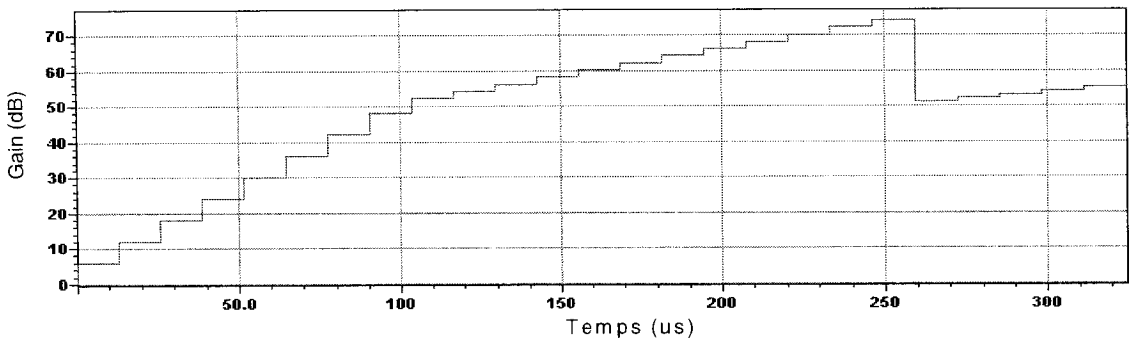


Figure 4-18 Fonction de transfert du modèle comportemental du TGC en fonction du temps pour des paramètres de programmation maximal.

La Figure 4-19 présente le signal « Original » amplifié par le modèle du TGC programmé avec des gains maximaux pour chaque champ. Nous constatons une saturation du signal après 90µs. La cause est liée à un gain qui est trop élevé a la fin de la zone proche et en zone moyenne. Le gain varie alors de 48dB (8 paliers avec un gain de 6dB) à 50dB (gain de départ de la zone moyenne). La figure montre aussi que l'amplification progresse effectivement par palier. Ceci implique donc l'apparition de distorsions dans le signal entre deux paliers de gain occasionnées par une amplification non linéaire, car dans le cas d'une amplification maximale la variation de gain entre deux paliers est de 6dB.

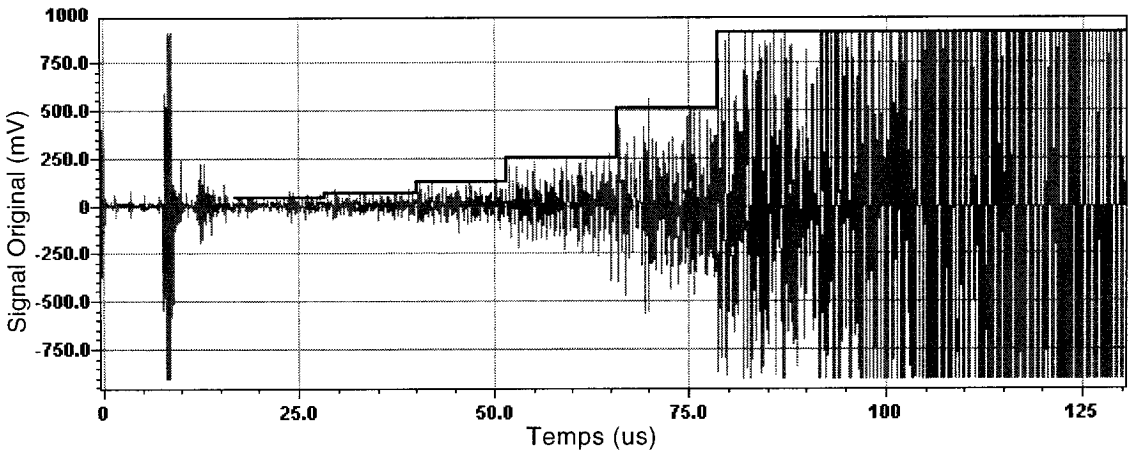


Figure 4-19 Signal « Original » amplifié par le modèle du TGC avec les paramètres de gain maximaux pour une profondeur de 10cm.

La Figure 4-20 présente le signal « Original » amplifié par le modèle du TGC avec les paramètres de valeurs moyennes. Dans ce cas-ci, l'impact du gain par palier sur l'ensemble du signal est beaucoup moins perceptible que dans le cas du TGC maximal. Par contre, il est à remarquer qu'il n'y a pas assez de gain au départ et qu'il y a une très grande transition d'amplitude à 104µs. Cette dernière s'explique par le fait que le gain à

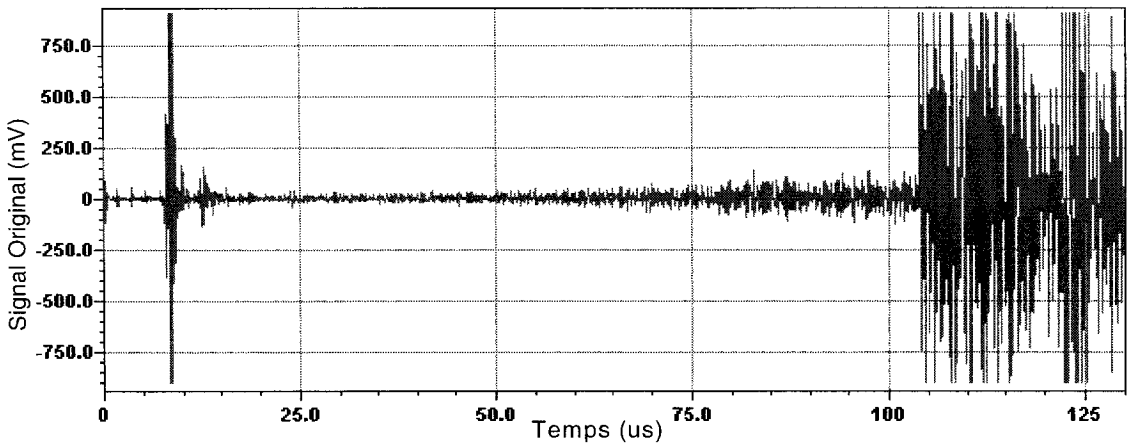


Figure 4-20 Signal « Original » amplifié par le modèle du TGC avec les paramètres moyens pour une profondeur de 10cm.

la fin de la zone proche est de 16dB (8 paliers avec un gain de 2dB) et que le gain qui débute la zone moyenne est de 50dB.

4.3.3 *Traitement numérique des signaux ultrasonores*

Cette section constitue la suite de la validation de tout le système. Le signal obtenu à la sortie du TGC est ensuite numérisé et envoyé à la détection d'enveloppe. La Figure 4-21 et la Figure 4-22 présentent les résultats de la détection d'enveloppe IQ pour les signaux du TGC moyen et maximal respectivement. Ces figures présentent une portion de 5cm afin de pouvoir comparer le signal avec celui obtenu à la Figure 4-4. La comparaison montre deux problèmes. Le premier est directement lié au TGC. Mis à part un certain décalage entre les signaux, les signaux sont identiques pour l'intervalle situé entre 12 et

30µs. Par contre, l'amplitude des signaux du TGC moyen et maximum est très faible (inférieure à 2.5mV) comparativement au signal de la Figure 4-4 qui varie entre 50 et 100mV. Le gain du TGC dans la zone proche est donc insuffisant. Le second problème vient du fait qu'il y a beaucoup de bruit, ce qui explique qu'il n'y a aucune ressemblance entre les signaux après 30µs. Dans ce dernier cas, la Figure 4-23 est plus appropriée pour voir l'impact du bruit. En fait, la comparaison entre cette figure et la Figure 4-3 montre la perte totale du signal situé entre 50 et 100µs.

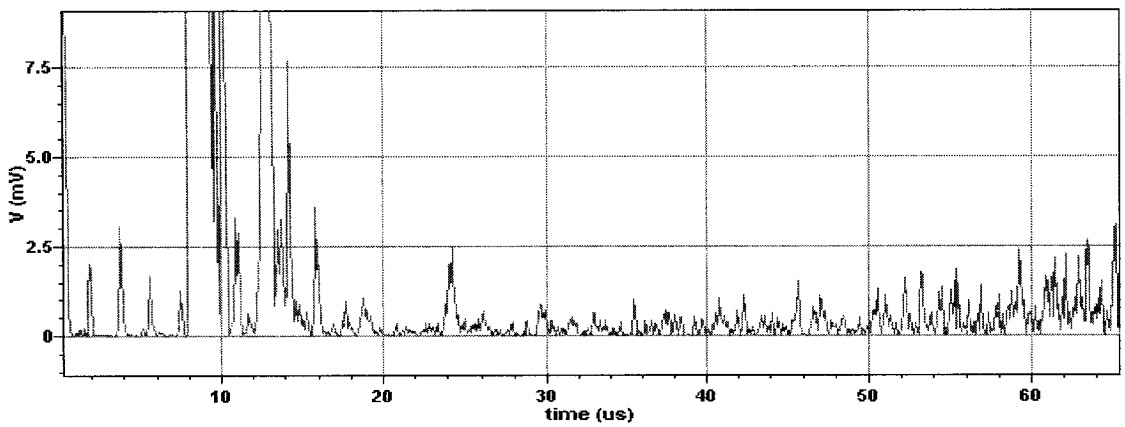


Figure 4-21 Graphique de l'enveloppe IQ du signal « Original » pour 5cm et amplifié par le TGC moyen.

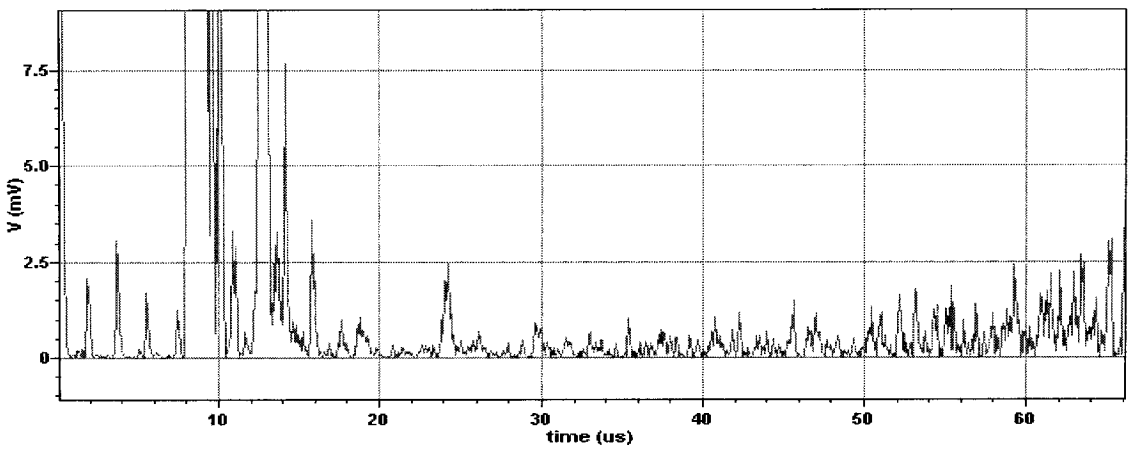


Figure 4-22 Graphique de l'enveloppe IQ du signal « Original » pour 5cm et amplifié par le TGC maximum.

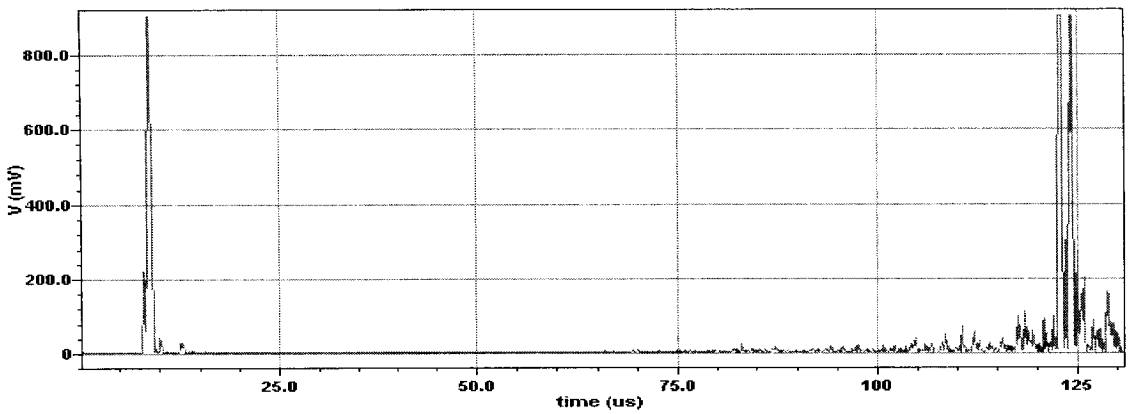


Figure 4-23 Graphique de l'enveloppe IQ du signal « Original » pour 10cm et amplifié par le TGC maximum.

La dernière étape est l'affichage du signal « Original ». La Figure 4-24 montre l'image obtenue. Les paramètres utilisés sont la modulation IQ, la compression linéaire, l'interpolation mixte et le choix du pixel le plus intense. En comparant cette image avec la Figure 4-10, il en ressort trois constatations. La première constatation est l'anneau blanc au bas de la Figure 4-24 représentant la saturation créée par le TGC. Ensuite à droite de la figure, nous remarquons l'émergence d'un faisceau de bruit, d'amplitude

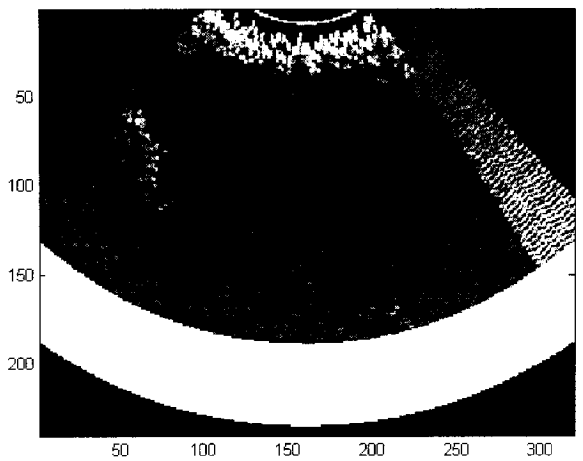


Figure 4-24 Résultat obtenu avec le signal « Original » et le TGC moyen pour une profondeur de 10cm.

constante, qui s'intensifie par étage de ton de gris. Ceci est encore dû au TGC. Enfin, la figure présente une légère similitude pour l'écho situé à gauche de l'image. Cet écho est très faible mais il ressemble, de par sa forme, à celui qui ce trouve à la Figure 4-10. Par contre, il n'y a aucune trace des images des cercles. Ceci est dû au fait que les échos ultrasonores provenant des cercles ont été perdus dans le bruit.

4.4 Validation et vérification des circuits réalisés par notre équipe

Cette section présente les résultats obtenus lors de la vérification et de la validation des circuits créés par notre équipe de recherche. De plus, les résultats des balayages du fantôme de la Figure 3-2b et de la Figure 3-3 sont présentés à la fin de cette section.

La Figure 4-25 présente le résultat de la simulation du circuit schématique du TLA avec pour entrée le signal de la Figure 4-14. La comparaison entre la Figure 4-16 et la Figure 4-25 montre qu'il y a saturation du signal. Par contre, il y a deux différences entre le modèle comportemental et le circuit schématique. Dans un premier temps, les deux signaux ont une plage de tension qui diffère d'environ 2V. Ceci est dû au fait que le modèle mathématique comportemental ne limite pas la tension de sortie à une plage d'environ 1.8V comme le fait le circuit schématique. Il suit plutôt la courbe de gain logarithmique du modèle comportemental. Si une limitation de tension de 1.8V était ajoutée au modèle comportemental, alors les résultats seraient presque identiques. Dans un deuxième temps, il y a une différence au niveau fréquentiel entre les deux signaux. En fait, l'oscillation présente à 10µs (Figure 4-16) n'est pas présente dans le résultat du circuit schématique. Ceci est causé par le fait que dans le modèle comportemental une tension élevée sera légèrement amplifiée tandis que dans le modèle schématique cette même tension sera limitée à 0.9V.

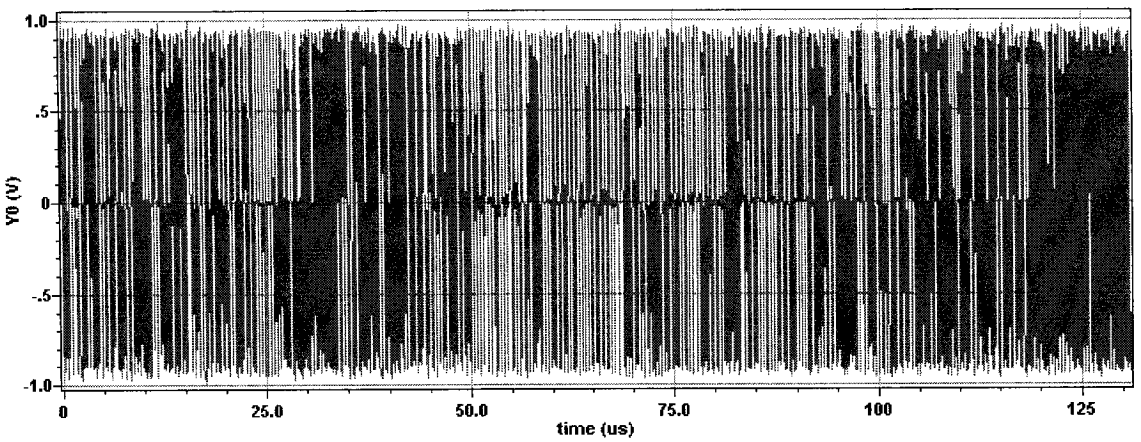


Figure 4-25 Résultat du circuit schématique du TLA.

La Figure 4-26 présente le résultat obtenu avec le circuit schématique du TGC calibré aux valeurs moyennes. La comparaison avec la Figure 4-20 montre que le modèle reflète le fonctionnement du TGC. Les deux signaux sont presque identiques.

Lors de la conception du TGC, il est important de tenir compte de l'intensité du signal qui sera émis par la sonde et de l'intervalle de tension générer par le transducteur à la réception de l'écho ultrasonore. En fait, dans le cas de la technologie 0.18µm CMOS, la plage de tension est limitée à environ 1.8V. Le problème vient du fait que pour la zone moyenne, soit à 8cm et plus, le gain débute minimalement à 44dB. Ceci implique qu'à cette distance la tension de l'écho reçu par le transducteur ne peut pas dépasser 5.6mV, sinon la sortie serait saturée. Le signal peut avoir une oscillation maximale de ±0.9V. La limite de tension se calcule alors de la façon suivante : 0.9/44dB = 0.9/158 = 5.6mV.

Un point important à noter est le fait que les données ultrasonores acquises ont été perçues dans un environnement constitué de gélatine et d'eau. Il existe donc une différence notable entre le coefficient d'atténuation dans l'eau qui est de 0.002 dB/cm et celui dans les tissus mous qui est de 0.6 dB/sm. Donc, lors de l'ajustement du gain du TGC il faudra que les données employées aient subi une atténuation semblable à un vrai environnement équivalent à celui du corps humain.

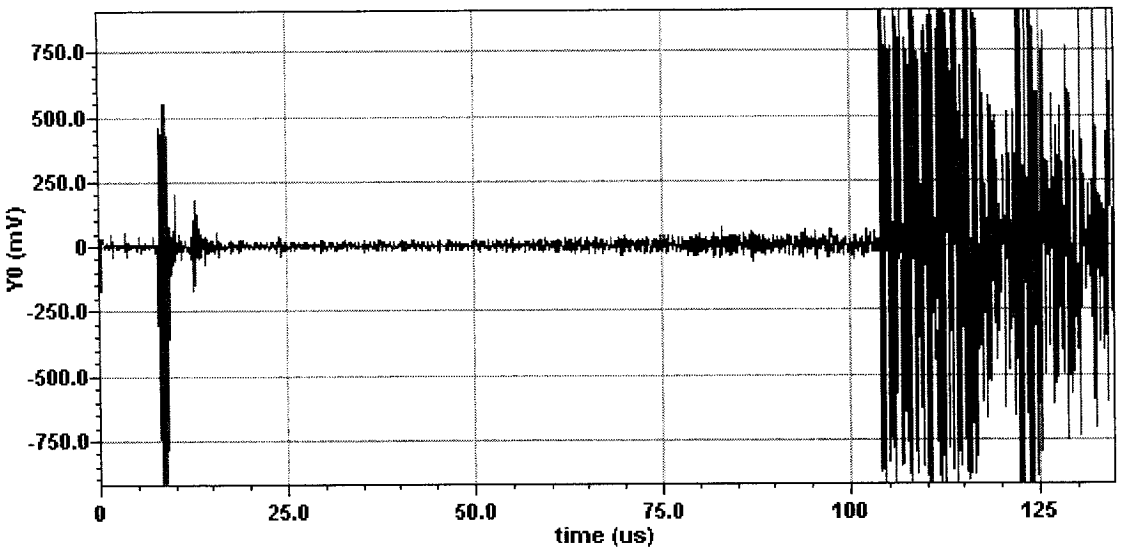


Figure 4-26 Signal « Original » amplifié par le circuit schématique du TGC avec les paramètres moyen pour une profondeur de 10cm.

La Figure 4-27 présente l'image adaptée qui a été traitée par le module de notre équipe de recherche. La figure utilisée possède 650 échantillons par ligne de balayage et est déjà convertie en données numériques compressées variant entre 0 et 40. C'est le seul moyen d'obtenir une image avec le module réalisé par notre équipe. En fait, l'utilisation directe du signal Ampli ou du signal provenant du CAN ne produit aucune image. Ceci est dû au fait que la détection d'enveloppe est non fonctionnelle. Par ailleurs, l'image obtenue en b est légèrement plus petite, ce qui s'explique par les paramètres de conversion utilisés par le module de notre équipe qui sont légèrement différents. De plus, ceci prouve que l'image est « beamformée ». En effet, le module de notre équipe effectue un « beamforming » lors du traitement de l'image. La comparaison de l'image obtenue avec celle présentée à la Figure 4-9 montre qu'il y a de légères différences dans l'image causées par le nombre d'échantillons employés, qui est dix fois moindre, et par la technique d'interpolation qui est différente.

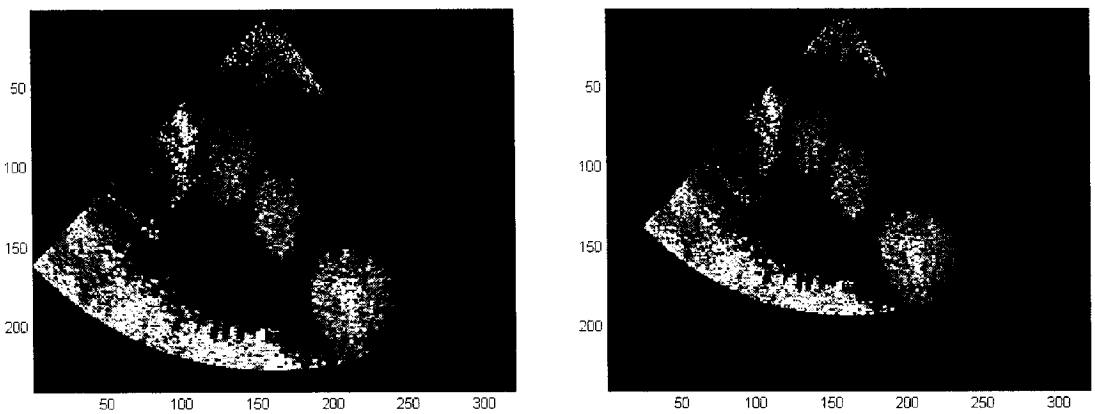
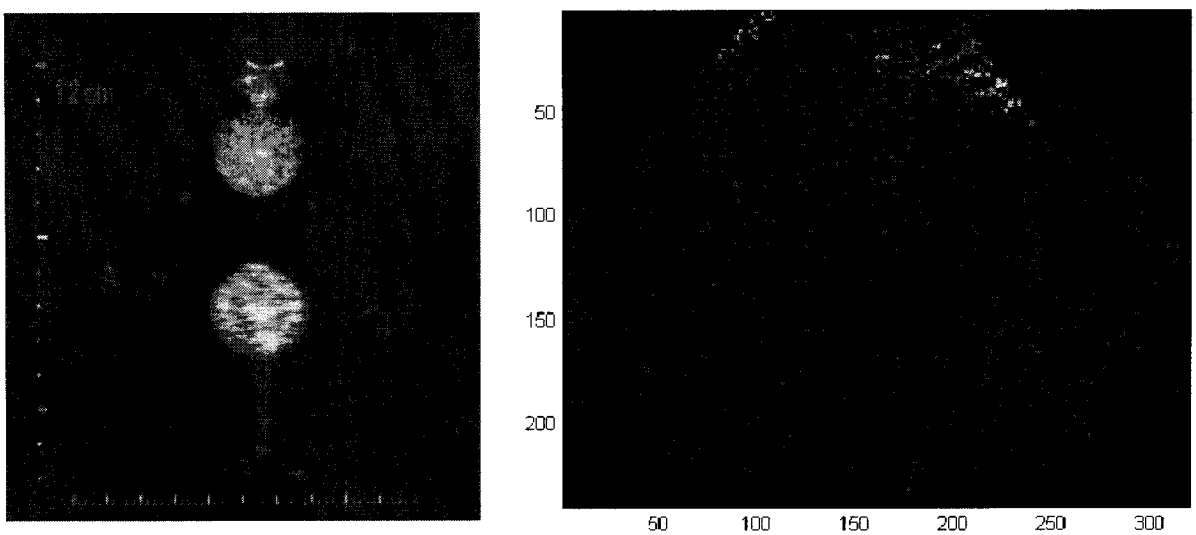


Figure 4-27 Image 1 de test pour le module numérique de l'équipe: a) Image utilisée pour la simulation et b) image résultant de la simulation.

Les images suivantes montrent d'autres exemples d'acquisition. La Figure 4-28 présente le résultat obtenue à partir du fantôme créé à la Figure 3-2b. L'image obtenue avec le modèle ne présente aucune forme. Ceci est dû à la présence du bruit lors de l'acquisition et au fait que le gain n'était pas assez élevé. Le résultat est le même pour les signaux « Original » et « Ampli », soit une image vide.



a) b)

Figure 4-28 Image 2 obtenue avec a) Scanview et b) le modèle

Le même constat de contamination par le bruit est fait pour la Figure 4-30 présentant une section de cœur. L'image est un peu plus visible puisque l'intensité de l'appareil était plus élevée lors de l'acquisition des données. L'image permet malgré tout de visualiser la région du ventricule gauche, de l'oreillette gauche ainsi qu'une petite partie des bicuspides.

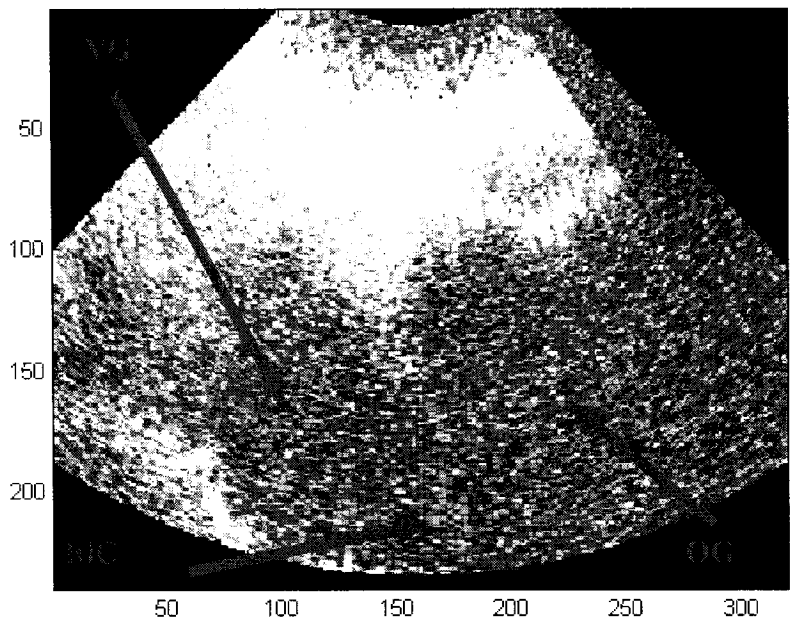


Figure 4-30 Image du cœur obtenu par la modélisation.

4.5 Autres aspects des résultats

Le Tableau 4-2 présente un résumé des temps de simulation pour l'ensemble des modules de l'environnement de tests. Les résultats sont compilés en fonction des profondeurs de 5cm et 10cm, ainsi que pour une profondeur de 10cm avec une décimation de 10 échantillons. La pertinence de ce tableau est dû au fait qu'il indique le temps requis pour

la simulation de chaque partie de l'environnement. Ceci permet donc de calculer le temps nécessaire à la validation d'un nouveau module. Ensuite, le total critique du tableau permet de constater que le temps est évidemment proportionnel au nombre de données utilisées. Juste à titre de comparaison, le module numérique conçu par notre équipe prenait en moyenne 3h pour simuler un balayage de 10cm avec 832 000 échantillons avant d'avoir l'affichage d'une image, et ce, sans le fonctionnement du détecteur d'enveloppe.

Tableau 4-2 Temps de simulation de l'environnement de tests

	Temps de simulation		
Quantité d'échantillon traité	832000	416000	83200
Profondeur	profondeur 10cm	profondeur 5cm	profondeur 10cm avec décimation
Acquisition	< 3 min	< 3 min	n/a
Extraction et segmentation	< 2 min	< 2 min	n/a
Numérique			
Détection et compression	< 2 min	< 2 min	n/a
Convertion et interpolation :			
Normal	5 min	4 min	1min
NNI et linéaire	11 min	9min	3 min
Mixte	16 min	13 min	5 min
Affichage			
Compression et affichage	2 min	2 min	< 1min
Analogique			
Modèle comprotemental	36 min	15 min	n/a
Modèle schématique	46 min	27 min	n/a
ADC	< 1 min	< 1min	n/a
Total critique	01h12	50 min	6 min

Au terme de cette section, la Figure 4-31 présente l'architecture de simulation employée lors de ce projet. Le but est de visualiser la structure des tests à effectuer lors de l'optimisation du système : par exemple, le fait que le TGC soit programmable et qu'il y ait en moyenne 3 techniques ou technologies à tester par étage. Il est alors possible de calculer un nombre total de simulations à réaliser pour tester tous les cas possibles rencontrés par le système. Ce nombre se chiffre au environ de 300 000 pour le modèle que nous proposons.

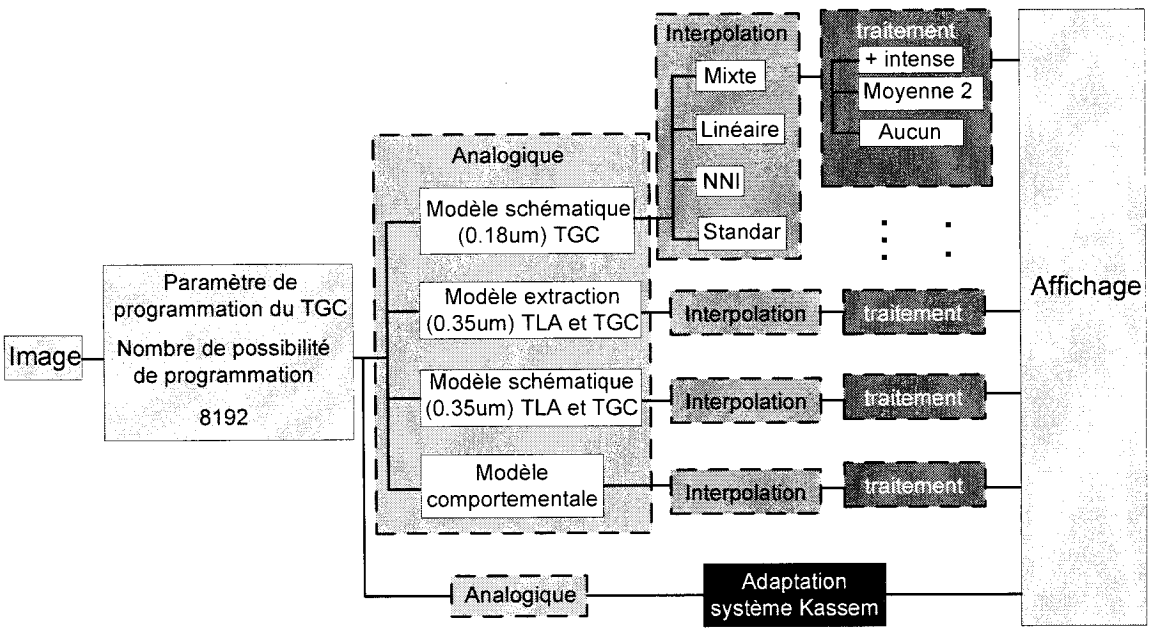


Figure 4-31 Architecture des combinaisons de simulation possibles

4.6 Autre développement du projet

Notre équipe travaille actuellement sur le développement d'éléments piézoélectriques intégrés sur des systèmes microélectromaniques (MEMS - Micro Electro Mecanichal Systems) pour exécuter les balayages électroniques et mécaniques de la sonde ultrasonore. En fait, notre système ressemblerait à la sonde présentée à la Figure 2-1. Avec le développement de la technologie de haut voltage (300V) sur puce, il est

maintenant possible de mettre au point des contrôleurs pour MEMS et d'exciter les éléments piézoélectriques sur une seule puce. Donc avec les connaissances acquises lors de cette maîtrise, nous avons conçu et réalisé un amplificateur opérationnel (contrôleur) de haut voltage sur puce avec la technologie CMOS HV 0.8 um de DALSA (CMC run ICKPMHDH). Tous les circuits du système étant maintenant concevables, le défi est donc de faire l'assemblage de tous les modules du système ultrasonore pour en extraire ses caractéristiques. Il faut alors adopter un environnement de tests propice à la vérification du fonctionnement et à l'étude des performances de ce système.

Conclusion

En somme, l'objectif visé par ce mémoire était de présenter l'importance de concevoir un environnement de tests permettant l'élaboration et la caractérisation d'un modèle de système ultrasonore portable. Les faits montrent que la miniaturisation des systèmes ultrasonores est une voie viable mais encore peu connue. Les avantages sont nombreux : technique non invasive, faible consommation d'énergie, portable, imagerie en temps réel et faible dépendance aux logiciels. Les progrès apportés par les logiciels de développement de systèmes sur puce ont amené de nouveaux outils qui doivent être testés afin de prendre conscience de leur efficacité et de leur utilité sur les systèmes mixtes. La modélisation devient maintenant une voie incontournable à la vérification et à la validation de systèmes entiers que l'on désire intégrer. Les innovations proposées par ce mémoire sont donc la possibilité de vérifier de nouveaux concepts, permettre l'optimisation facile de systèmes ultrasonores, amener une analyse rigoureuse des paramètres dont doit tenir compte le système et permettre de faire le traitement complet d'une image échographique en débutant par le mode A jusqu'au mode B pour concrétiser le développement de nouveaux concepts. De plus, avec les connaissances acquisent, nous avons réalisé un amplificateur de haut voltage (0 – 300V) à faible consommation d'énergie sur puce. Ce projet ne fait pas partie de ce mémoire mais montre la possibilité d'utiliser des transducteurs entièrement intégrés. La voie de la technologie des MEMS dans la conception des futures sondes est donc parfaitement ouverte.

Les travaux antérieurs de notre équipe de recherche proposaient un système de réception analogique et numérique ayant fait l'objet de divers tests de fonctionnalité. Par contre, le modèle que nous avons proposé a permis de déceler quelques lacunes : l'amplification inadaptée du TLA, la distorsion occasionnée par le TGC, la détection d'enveloppe non utilisée et l'interpolation qui manque de précision. Aussi les tests n'étaient pas rigoureux face à l'évaluation du fonctionnement du système dans un contexte réel d'imagerie

ultrasonore. À la lumière de nos recherches, nous avons donc réussi à proposer une méthode de tests qui tient compte des différents facteurs pouvant influencer le fonctionnement ou la précision des appareils ultrasonores.

L'environnement de tests (modèle) créé reflète l'ensemble de cette méthode. La conception d'un banc de tests pour faire l'acquisition de signaux échographiques en mode A à partir de fantômes permet le contrôle des variables constituant l'environnement. Le milieu est donc homogène, paramétré, les objets balayés sont visualisables, ce qui fournit des images paramétrés. Le banc d'essais analogique permet de faire des expérimentations à variables contrôlées et par groupe de contrôle. Le but est de fournir les mêmes données au modèle comportemental, au circuit schématique et à l'extraction des dessins des masques. L'importance de cette étape est de s'assurer d'une erreur de fonctionnement raisonnable. L'accent peut donc être mis sur la conception de circuits constituant le système recherché. Le banc d'essais numérique permet le même type d'expérimentation que le banc analogique. Par contre, l'accent est plutôt porté sur l'optimisation des techniques de traitement de l'information. Chaque module constituant le système numérique peut être vérifié individuellement et incorporé, par la suite, au système global. De plus, le fait de pouvoir implémenter tout le concept numérique dans un FPGA rend la vérification fonctionnelle réelle beaucoup plus flexible, rapide et peu dispendieuse. Enfin, l'affichage permet de concrétiser les efforts mis dans l'optimisation du système global en fournissant un résultat visuel de tout le traitement effectué sur le signal ultrasonore de départ. Il fournit un bon indice qualitatif et quantitatif sur les résultats obtenus. Par ailleurs, la plus grande qualité de cette modélisation est que l'information nécessaire à la vérification d'un module peut être échantillonnée à n'importe quel étage du système complet et réutilisée. Nous avons donc atteint les objectifs suivants :

1. Les outils que nous avons développés dans ce mémoire permettent de caractériser chaque module conçu pour le système échographique en développement dans notre groupe.

2. La vitesse de vérification a effectivement été accrue, la méthode de programmation en Verilog et Verilog-A ainsi que les données réelles utilisées permettent non pas d'optimiser mais de voir plus facilement l'impact d'un nouveau module dans le système de traitement du signal. Enfin, les données sont comparables entre elles puisque le signal est le même pour chaque module et que les données peuvent être analysées indépendamment. L'impact sur le système est immédiat.

3. Les signaux ultrasonores acquis proviennent directement de piézoélectriques. La particularité de pouvoir faire l'acquisition et la segmentation de ces signaux ultrasonores réside dans le fait que ces signaux peuvent être analysés de multiples façons. En effet, il est possible de vérifier la qualité (facteur Q) du signal émis par le transducteur, la fréquence, la variation de la fréquence et de l'amplitude en fonction de la profondeur des échos reçus, la fonctionnalité des modules du front-end et la reconstruction du signal réel. Ainsi, on évalue vraiment la performance réelle de l'étage de réception.

4. Nous avons pu localiser des erreurs au niveau du système proposé auparavant par notre équipe. Nous avons apporté certaines modifications qui ont permis au système de fonctionner convenablement autant au niveau numérique qu'analogique pour une fréquence de 5MHz.

5. L'optimisation du système est rendue possible au sens où il est désormais facile d'améliorer une partie unique du système global. La méthodologie est éclairée par l'élaboration du modèle du processus de traitement du signal ultrasonore, ce qui laisse place à l'innovation au niveau des différents modules qui composent le système. De plus, le fait de voir l'impact immédiat permet aussi de corriger plus rapidement les erreurs, autant au niveau du modèle comportemental, schématique que du dessin des masques. La partie analogique peut aussi être utilisée dans tous les procédés de fabrication (0.35µm, 0.18µm, etc.).

Recommandations sur le projet

Il est primordial de réduire considérablement le bruit à l'entrée de l'étage de réception à l'aide d'amplificateurs de faible bruit ou de filtres. Par ailleurs, nous pouvons déjà nous attendre à une réduction considérable du bruit par le fait que le transducteur et l'amplificateur seront sur un même ASIC [19].

L'architecture du TGC élaborée par l'équipe possède deux lacunes. La première concerne la résolution du signal amplifié. En effet, puisque le gain augmente par palier, l'information provenant des échos à proximité des discontinuités de gain est alors distordue. L'amplitude étant directement reliée à l'intensité du pixel, la discontinuité engendre donc des images avec des zones d'intensité non constantes. Ceci est d'autant plus vrai que l'amplification entre chaque palier est grande. La seconde lacune est le manque d'information concernant le lien entre l'amplification que le TGC fournit et la puissance de la sonde utilisée. Il est présumé que l'intensité d'un écho varie entre 5μV et 500mV en général. Par contre, lors de la conception d'un TGC, il faut déterminer la puissance fournie par le transducteur au milieu dans lequel il se propage. Il s'agit simplement de calculer le meilleur rapport entre la plage de tension disponible et l'amplitude de la tension de l'écho reçu par le transducteur en fonction de la profondeur. Ceci permettra de déterminer la puissance que doit fournir la sonde ou la variation de gain du TGC nécessaire afin de ne pas saturer l'étage de réception.

Le gain de 100dB du TLA n'est vraiment pas approprié à l'entrée de l'étage de réception. D'après les tests, c'est un amplificateur de bruit en plus d'engendrer un facteur d'erreur au niveau de l'information que peut fournir l'écho. En effet, la variation logarithmique de l'amplitude vient déformer l'information fournie par l'écho, car l'amplitude fournie de l'information sur la nature de la structure rencontrée. D'après les faits, l'atténuation générale suit plutôt une courbe exponentielle et non pas logarithmique comme le montre l'équation **(1.7)**. Il faut noter que la structure n'est pas simplement détectée mais aussi

reconstituée. Il faut donc garder les subtilités fournies par les variations de tension du signal ultrasonore reçu.

L'analyse du programme VHDL du DBF et le fait d'avoir corroboré les résultats obtenus par notre équipe montrent que le module est bien fonctionnel. La seule amélioration qui peut être faite est liée à l'erreur du pas numérique lors de la synchronisation des signaux. Il s'agit alors de connaître les paramètres qui entrent en ligne de compte pour la synchronisation des signaux : la résolution de synchronisation qui est liée à la vitesse de l'horloge, la distance entre les transducteurs linéaires et l'angle du balayage effectué. Ceci lie donc le délai calculé et le délai généré par les composantes numériques afin de synchroniser parfaitement les signaux ultrasonores.

Les résultats obtenus avec le DSP montrent qu'il y a plusieurs possibilités d'optimisation de la modulation, de la démodulation ou simplement du filtrage pour la détection d'enveloppe. Les bons résultats obtenus avec le filtre IIR demande que le sujet soit exploité autant pour la modulation IQ dont les performances de signal sur bruit sont très bonnes.

En ce qui concerne la compression vidéo, le sujet est entièrement ouvert. Il s'agit maintenant de déterminer la meilleure méthode à employer. Les méthodes par logarithme et par simple conversion de N bits à 6 bits utilisés par notre équipe montrent qu'il reste du travail à faire à ce niveau. Il faut rappeler que la compression génère aussi des erreurs de résolutions selon la méthode adoptée comme il est possible de le voir avec les résultats obtenus.

La conversion du balayage en données cartésiennes fait place à plusieurs techniques selon le dilemme entre la mémoire et l'espace sur puce disponible. Il est possible d'utiliser la mémoire pour le calcul des positions des pixels avec des « look up table » ou encore de faire la conception d'un algorithme séquentiel qui positionne les intensités de pixel. Il

faut aussi choisir la méthode de sélection des intensités de pixel qui sera utilisée. Dans ce document les deux méthodes employées étaient le pixel le plus intense est gardé et la moyenne de 2.

L'interpolation est très importante du point de vue de la qualité d'image voulue. Plusieurs techniques existent et l'impact sur la qualité de l'image a pu être constaté. Les observations effectuées concernaient surtout le temps, les calculs et la mémoire disponible. Par exemple, la méthode d'interpolation proposée par notre équipe montre que le nombre de vecteurs nécessaires pour une bonne image de 320 sur 240 est de 512 vecteurs. Par contre, ceci représente plusieurs calculs de positions et d'intensités. Il s'ensuit une exploitation très redondante des données de positions et d'intensités par rapport à la technique de multizones présentée à la Figure 2-4.

Finalement, dans un futur rapproché il faudrait :

1) Établir une méthodologie de choix de tests vu l'ampleur des solutions possibles pour vérifier les modules;
2) Investir du temps afin d'examiner les solutions optimales pour chaque étage de traitement du signal (analogique et numérique);
3) Générer un système entièrement conçu en Verilog-AMS avec system Verilog. (un seul langage, possibilité d'accroître la vitesse d'exécution et moins de manipulation de données);
4) Développement à faire au niveau du balayage employé (électronique). Potentiel énorme en économie de temps et en résolution.

BIBLIOGRAPHIE

Livres et articles

[1] ASHFAQ M., MIENKINA M.P., ERMERT H., « Estimation of time of flight for ultrasonic reflex-transmission tomography with active contour models », IEEE International Ultrasonics, Ferroelectrics and Frequency Control Conference, p. 1433-1436, 2004.

[2] BADAWY W., « System-on-chip : Issue, challenges and trends », Can. J. Elect. & Comp. Eng., Vol 26, No 3/4, p. 85-90, july/oct. 2001.

[3] BEAUCHAMPS-PARENT, A. Alarme de conditionnement ultrasonore programmable pour le traitement de l'énuresie. 1999. 197 p. Mémoire de maîtrise, Département de génie électrique. École Polytechnique de Montréal.

[4] BOTROS N., « A high speed data acquisition and analysis system for ultrasonic energy measurements », IEEE Transactions on Instrumentation and Measurement, Vol. 37, Issue 4, p. 515-518, dec. 1988.

[5] CAROTENUTO R., CALIANO G., CARONTI A., « Very fast scanning probe for ophthalmic echography using an ultrasound motor », IEEE Ultrasonics Symposium, Vol. 2, p. 1310-1313, 2004.

[6] CHEN Q.X., PAYNE P.A., « Industrial applications of piezoelectric polymer transducers », Meas. Sci,. Technol. 6, p. 249-267, 1995

[7] CHEN W.H., CAO P.J., MAIONE E., RITTER T., SHUNG K.K., « Optimization of pulse transmission in a high-frenquency ultrasound imaging system », IEEE Ultrasonics Symposium, p. 995-998, 2001.

[8] CHEBLI Robert. Étage de réception intégré d'un échographe portable. 2002. 161 p. Mémoire de maîtrise, Département de génie électrique. École polytechnique de Montréal.

[9] CROKER J.M. Handbook of acoustics. 1th ed. John Wiley & Sons, inc., 1998. 1461 p.

[10] DEMIRCI U., ORALKAN Ö., JOHNSON A. J., ERGUN S. A., KARAMAN M., KHURI-YARUB T. B., « Capacitive micromachined ultrasonic transducer arrays for medical imaging : Experimental results ». IEEE Ultrasonics Symposium, p. 957-960, 2001.

[11] FITZPATRICK D., MILLER I., « Analog behavioral modeling with the verilog-a language », Kluwer Academic Publishers, USA, p.210, 1999.

[12] FRITSH C., PARRILLA M., MARTINEZ O., JEMENEZ D., « A multirate scan conversion method », Ultrasonics, 38, p.1779-182, 2000.

[13] GREENBERG N.L., SARACINO G., BOROWSKI A.G., THOMAS J.D., « Wireless echocardiographic image acquisition and review », IEEE Computer in Cardiology, 31, p. 741-743, sept. 2004.

[14] HWANG J.J., QUISTGAARD J., SOUQUET J., CRUM A.L., « Portable ultrasound device for battlefield trauma », IEEE Ultrasonics Symposium, Vol. 2, p. 1663-1667,1998.

[15] HUANG Q.H., ZHEN Y.P., LU M.H., CHI Z.R., « Development of a portable 3D ultrasound imaging system for musculoskeletal tissues », Ultrasonics, 43, p. 153-163, 2005.

[16] JOHNSON J., ORALKAN Ö., DEMIRCI U., ERGUN S., KARAMAN M., KHURI-YAKUB P., « Medical imaging using capacitive micromachined ultrasonic transducer arrays », Ultrasonics, 40, p. 471-476, may 2002.

[17] KASSEM. A. Technique de conception système sur une puce dédiée à l'imagerie par ultrasons. 2004. 149 p. Thèse de doctorat, Département de génie électrique, École polytechnique de Montréal.

[18] KASSEM A., SAWAN M., BOUKADOUM M. « A new digital scan conversion architecture for ultrasonic imaging system », World Scientific, Journal of Circuits, Systems and Computers, Vol. 14, No 2, p. 367-382, avril 2005.

[19] KAVIANI K., ORALKAN Ö., KHURI-YARUB P., WOOLEY B.A., « A multichannel pipeline analog-to-digital converter for an integrated 3D Ultrasound Imaging System », IEEE Journal of Solid-State Circuit, Vol. 38, No. 7, july 2003.

[20] KENDERT K., CHANG H., JEFFERIES D., LAMANT G., MALAVASI E. SENDIG F., « Design of mixed-signal systems-on-a-chip », IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 19, No. 12, dec. 2000.

[21] KIM Y., KIM H. J., BASOGLU C., WINTER C. T., « Programmable ultrasound imaging using multimedia technologie : A next-generation ultrasound machine », IEEE Transaction on Information Technologie in Biomedicine, vol. 1, No 1, march 1997.

[22] LAY L.L., CAREY S.J., HATFIELD J.V., « Pre-amplifier arrays for intra-oral ultrasound probe receiving electronics », IEEE International Ultrasonics, Ferroelectrics and Frequency Control Conference, Vol. 3, p. 1753-1756, aug. 2004.

[23] LO S.-C. B., LIU C. C., FREEDMAN T. M., KULA J., LASSER B., LASSER E. M., WANG Y., « Transmission and reflective ultrasound image using PE-CMOS sensor array », SPIE Proc. Med. Img., Vol. 5750, 2005.

[24] LOUKUSA V., « Behavioral test generation modeling approach for mixed-signal IC verification », Microelectronics Journal, 34, p. 907-912, april 2003.

[25] MAL A., FENG F., KABO M., WANG J., « Interaction of high intensity focused ultrasound with biological materials », SPIE's NDE and Smart Strutures Symp., Paper 4702-40, San Diego, march 2002.

[26] MATSUO I., YANO M., « An echolocation model for the restoration of an acoustic image from a single-emission echo », J. Acoust. Soc. Am., Vol. 116, No 6, dec. 2004.

[27] McINTOSH J.S., HUTCHINS D.A., BILLSON D.R., ROBERTSON T.J., NOBLE R.A., JONES A.D.R., « The characterization of capacitive micromachined ultrasonic transducers in air », Ultrasonics, 40, p. 477-483, may 2002.

[28] MILLER I., CASSAGNES T., « Verilog-a and verilog-ams provides a new dimension in modeling and simulation », Proc. IEEE International Caracas Conference on Devices, Circuits and Systems, p. 1-6, march 2000.

[29] MOHAMED HADI. W. Restauration de signaux radiofrequences obtenus a partir des composantes IQ: Effets sur l'elastogramme. 2004. 143 p. Mémoire de maîtrise, Département de génie électrique. École polytechnique de Montréal.

[30] PARK S., « Principles of sigma-delta modulation for analog-to-digital converters », MOTOROLA, USA, APR8/D Rev.1, 1990 p.

[31] PECHEUX F., LALLEMENT C., VACHOUX A., « VHDL-AMS and Verilog-AMS as alternative hardware description language for efficient modeling of multidiscipline systems », IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 2, p. 204-225, feb. 2005.

[32] RADULESCU E.G., LEWIN P.A., NOWICKI A., SHANKAR P.M., « Performance evaluation of wideband ultrasound hydrophone probes beyond 20 MHz », Proc. IEEE 28th Annual Northeast Bioengineering Conference, Philadelphia, PA, p. 249-250, april 2002.

[33] SAIJO Y., NITTA S.I., KOBAYASHI K., ARAI H., NEMOTO Y., « Development of an ultra-portable echo device connected to USB port », Ultrasonics, 42, p. 699-703, 2004.

[34] SAWAN M., CHEBLI R., KASSEM A., « Integrated front-end receiver for a portable ultrasonic system », Kluwer Academic Publishers, Analog Integrated Circuits and Signal Processing, 36, p. 57-67, 2003.

[35] SHUNG K.K., ZIPPARO R.T., « Ultrasonic transducers and arrays », IEEE Engineering in Medicine and Biology, Vol. 15, No. 6, p. 20-30, 1996.

[36] SMITH-LEVITIN M., BLICKSTEIN I., ALBRECHT-SHACH A.A., GOLDMAN R.D., GUREWITSCH E., STRELTZOFF J., CHERVENAK F.A., « Quantitative assessment of gray-level perception : obervers' accuracy is dependent on density differences », Ultrasound Obstet. Gynecol., 10, p. 346-349, 1997.

[37] VESETAS R., MANZIE G., « AMI : A 3-D imaging sonar for mine identification in turbid waters », MTS/IEEE Conference and Exhibition OCEANS, Vol. 1, p. 12-21, nov. 2001.

[38] WELLS P.N.T., « Ultrasonic imaging of the human body », Rep. Prog. Phys., 62, p. 671-722, nov. 1998.

[39] WELLS P.N.T., « Ultrasonics in medicine and biology », Phys. Med. Biol., Vol. 22, No. 4, p. 629-669, jan. 1977.

[40] WYGANT O. I., ZHUANG X., YEH T. D., NIKOOZADEH A., ORALKAN O., ERGUN S. A., KARAMAN M., KHURI-YAKUB T. B., « Integrated ultrasonic imaging systems based on CMUT arrays : Recent progress », IEEE International Ultrasonics, Ferroelectrics and Frequency Control Conference, p. 391-394, 2004

Sites web

[41] ADVANCED IMAGING ASSOCIATES, http://www.advancedimagingassoc.com/services.html, (Consulté en sept. 2005)

[42] BIOSOUND ESAOTE, http://www.biosound.com, (Consulté en oct. 2004)

[43] CADENCE DESIGN SYSTEM, « Affirma Verilog-a language reference », http://ece-www.colorado.edu/~ecen5007/cadence/files/veriaref.pdf, USA, Production Version 4.4.6, p. 413, July 2001. (Consulté en sept. 2004).

[44] CHISON, http://www.chison.com.cn/, (Consulté en oct. 2004)

[45] DORAN J.S., 2005, « An introduction to Medical Imaging », http://www.ph.surrey.ac.uk /~phs3sd/, (consulté en décembre 2004)

[46] FUKUDA DENSHI, http://www.fukuda.com/, (Consulté en oct. 2004)

[47] FULTON SCHOOLS, http://fulton.edzone.net/winkler/winkler_menu_0506.html, (Consulté en sept. 2005)

[48] GE HEALTCARE, http://www.gehealthcare.com/usen/ultrasound/index.html, (Consulté en oct. 2004)

[49] HITACHI. http://www.hitachi.us/Apps/hitachicom/content.jsp?page=index.html&&_path=jsp/hitachi/forbus/medical/, (Consulté en oct. 2004)

[50] KENT MEDICAL IMAGING, http://www.kentmedicalimaging.co.uk/vascular.htm, (Consulté en sept 2005)

[51] KONTRON MEDICAL, http://kontron2004.veloce-it.net/, (Consulté en oct. 2004)

[52] LAWSON IMAGING, www.lawsonimaging.ca, (Consulté en sept. 2005)

[53] L.I.M. Université de Renne CT rayon X, http://www.med.univ-rennes1.fr/cerf/edicerf/BASES/BA001_idx.html (Consulté en oct. 2004)

[54] L.I.M. Université de Renne IRM, http://www.med.univ-rennes1.fr/cerf/edicerf/BASES/BA004_idx.html (Consulté en oct. 2004)

[55] L.I.M. Université de Renne IUS, http://www.med.univ-rennes1.fr/cerf/edicerf/BASES/BA003_idx.html (Consulté en oct. 2004)

[56] MEDISON USA, http://www.medisonusa.com/, (Consulté en oct. 2004)

[57] PHILIPS MEDICAL, http://www.medical.philips.com/main/products/ultrasound/, (Consulté en oct. 2004)

[58] SCANVIEW, http://www.scanview.ca, (Consulté en oct. 2004)

[59] SEER's TRAINING WEBSITE, http://training.seer.cancer.gov/module_abstracting/unit03_sec02_part03_computerized_tomo.html, (Consulté en sep. 2005)

[60] SHIMADZU MEDICAL, http://www1.shimadzu.com/products/medical/ultra.html, (Consulté en oct. 2004)

[61] SIEMENS MEDICAL, http://www.medical.siemens.com/webapp/wcs/stores/servlet/CategoryDisplay?storeId=10001&catalogId=-1&langId=-1&categoryId=12761&catTree=100001,12805,12761&level=0, (Consulté en oct. 2004)

[62] SIUI, http://www.siui.com/english/enidex.html, (Consulté en oct. 2004)

[63] SMITH D.J., « VHDL & Verilog compared & contrasted plus modeled example written in VHDL, Verilog and C », http://www.angelfire.com/in/rajesh52/verilogvhdl.html, (Consulté en sep. 2004).

[64] SONOSITE, http://www.sonosite.com/, (Consulté en oct. 2004)

[65] TERASON TERATECH, http://www.terason.com/, (Consulté en oct. 2004)

[66] TEXAS UNIVERSITY AT AUSTIN, http://www.bme.utexas.edu/ugrad/UGLab/slides.html, (Consulté en oct. 2005)

[67] TOSHIBA MEDICAL, http://www.medical.toshiba.com/clinical/radiology/radiology-2-5.htm, (Consulté en oct. 2004)

[68] ULTRASONIX MEDICAL, http://www.ultrasonix.com/, (Consulté en oct. 2004)

[69] WISCONSIN UNIVERSITY MADISON, http://zoot.radiology.wisc.edu/~block/Med_Gallery/ct_heart.html, (Consulté en sept 2005)

[70] WOO J., « A short history of the development of ultrasound in obstetrics ans gynecology : part 1-3 », http://www.ob-ultrasound.net/history1.html, (consulté en novembre 2004).

[71] ZONARE MEDICAL SYSTEM, http://www.zonare.com/opencms/Home, (Consulté en oct. 2004)

Réalisations

Présentation de mon projet à la conférence de l'AITS (Association des Industriels de la Technologie de la Santé), Montréal, Mars 2006.

VEILLEUX V.N., SAWAN M. « Modélisation d'une architecture générique d'un système ultrasonore embarqué », TAISA, 2006. (En attente d'acceptation)

ANNEXE

MODELSIM

Banc d'essai d'un module quelconque

```
//**********************************************************************
//  Module de test
//  clk a 50 MHz
//**********************************************************************

module TB_extraction;

reg clk;

initial begin
        clk = 1'b0; // initialisation de la clk a 0
end

always begin
        #10 clk = ~clk; // oscillation de la clk
end

// Appel du bloc que l'on simule

Extraction test1(clk);

Endmodule
```

Extraction du signal

```
//**********************************************************************
// Ce programme permet de faire la saisie des signaux constituant les 128 lignes de scan.
// Il extrait la secance utile du signal complet provenant de l'acquisitionneur de donner.
//
// Nom: Nicolas V.Veilleux
// Date: 2005
//
//**********************************************************************

`define Temps_Depart 0                  // En nombre d'echantillon saute
`define Temps_Fin 9750                  // En nombre d'echantillon saute
`define Debut_Scan 17                   // En nombre de front haut de Fire
`define Seuil_Fire_Bas 0                // En tension
`define Seuil_Fire_Haut 0.14
`define Seuil_Start 0.01                // En tension
`define NULL 0
```

```
`define EOF 32'hFFFF_FFFF
`define Fin_Scan 271                    // En nombre de front haut de Fire
`define F_Sample 10


module Extraction(Clk);

        input Clk;

  integer Filer_Start;              // Ouverture du fichier comptenant le signal du debut du balayage
        integer Filer_Fire;         // Ouverture du fichier des signaux indiquant le debut d'une ligne
        integer Filer_Echo;         // Ouverture de fichier comptenant les echos


        integer Cond_Ouvert_Start;          // Condition d'ouverture du fichier lecture
        integer Cond_Ouvert_Fire;
        integer Cond_Ouvert_Echo;
        integer C_Start;                    // C variable de lecture
        integer C_Fire;
        integer C_Echo;
        integer Filew;                      // Ouverture de fichier d'ecriture
        integer Cond_Ouvert_w;              // Condition d'ouverture du fichier d'ecriture
        integer Nb_Sample;                  // Nombre d'echantillon
        integer T_Nb_Sample;
        integer Front_Fire;
        integer T_Front_Fire;
        integer Nb_Front_Fire;
        integer T_Nb_Front_Fire;
        integer Detect_Start;
        integer Front_Start;
        integer T_Detect_Start;
        integer T_Front_Start;
        integer compteur;
        integer T_compteur;

        real Decimal_Start;                 // Registre de la donnee reel lu
        real Decimal_Fire;
        real Decimal_Echo;

 initial begin

 // Ouverture des differents fichier a lire

            Filer_Start = $fopen("/home/veilleux/DonneeTest/DATA/file_start.txt", "r");
    if (Filer_Start == `NULL)
            Cond_Ouvert_Start = 0;
    else
            Cond_Ouvert_Start = 1;
    $display("Ouverture Start :", Cond_Ouvert_Start);


        Filer_Fire = $fopen("/home/veilleux/DonneeTest/DATA/file_fire.txt", "r");
        if (Filer_Fire == `NULL)
```

```
                Cond_Ouvert_Fire = 0;
        else
                Cond_Ouvert_Fire = 1;
        $display("Ouverture Fire :", Cond_Ouvert_Fire);

    Filer_Echo = $fopen("/home/veilleux/DonneeTest/DATA/file_echo.txt", "r");
    if (Filer_Echo == `NULL)
            Cond_Ouvert_Echo = 0;
    else
            Cond_Ouvert_Echo = 1;
    $display("Ouverture Echo :", Cond_Ouvert_Echo);

// Ouverture du fichier d'ecriture

        Filew = $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/echo_traite.txt", "w");
        if (Filew == `NULL)
                Cond_Ouvert_w = 0;
        else
                Cond_Ouvert_w = 1;

        $display(" Ouverture Echo Traite :", Cond_Ouvert_w);

// Initialisation des variables

 Nb_Sample = 0;
 T_Nb_Sample = 0;
 Front_Fire = 0;
 T_Front_Fire = 0;
 Nb_Front_Fire = 0;
 T_Nb_Front_Fire = 0;
 Detect_Start = 0;
 Front_Start = 0;
 T_Detect_Start = 0;
 T_Front_Start =0;
 compteur = 0;
 T_compteur = 0;

 C_Start = $fscanf(Filer_Start, " %f \n ", Decimal_Start);
 C_Fire = $fscanf(Filer_Fire, " %f \n ", Decimal_Fire);
 C_Echo = $fscanf(Filer_Echo, " %f \n ", Decimal_Echo);

 end

 always @(posedge Clk) begin

          if((C_Start || C_Fire || C_Echo) != `EOF)
           begin


 // Lecture des fichiers

                  if(Detect_Start == 0)
                                  C_Start = $fscanf(Filer_Start, " %f \n ", Decimal_Start);
```

```
    C_Fire = $fscanf(Filer_Fire, " %f \n ", Decimal_Fire);
    C_Echo = $fscanf(Filer_Echo, " %f \n ", Decimal_Echo);

            if(Decimal_Fire <= `Seuil_Fire_Bas)
              T_Front_Fire = 1;

    Detect_Start = T_Detect_Start;
    Front_Start = T_Front_Start;
            Nb_Sample = T_Nb_Sample;
            Front_Fire = T_Front_Fire;
            Nb_Front_Fire = T_Nb_Front_Fire;
            T_compteur = compteur;
           end
          end

always @(negedge Clk) begin


          if( (C_Start || C_Fire || C_Echo) != `EOF & Nb_Front_Fire <= `Fin_Scan)
           begin

// Detection signal Start

            if (Decimal_Start <= `Seuil_Start)
              begin
                T_Front_Start = 1;
                T_Detect_Start = 1;
              end

// Detection signal Fire

            if (Decimal_Fire > `Seuil_Fire_Haut & Front_Fire == 1 & Front_Start == 1)
              begin
                T_Nb_Front_Fire = Nb_Front_Fire + 1;
        T_Front_Fire = 0;
             end

// Ecriture des donnees valides

            if (T_Nb_Front_Fire >= `Debut_Scan & T_Nb_Front_Fire <= `Fin_Scan)
             begin

                if ((T_Nb_Front_Fire % 2) != 0 & Decimal_Fire > `Seuil_Fire_Haut & Front_Fire == 1 &
Front_Start == 1)
                   begin
                    T_Nb_Sample = 0;
                   end


                else
                  T_Nb_Sample = Nb_Sample + 1;
```

```
            if(Nb_Sample >= `Temps_Depart & Nb_Sample <= `Temps_Fin - 1 &
(Nb_Sample%`F_Sample) == 0)
              begin
              $fdisplayb(Filew, Decimal_Echo);

              compteur = T_compteur + 1;
              end
          end
        if(Nb_Front_Fire == `Fin_Scan)
          $display(" total sample :", compteur);
         end
end
endmodule
```

Segmentation du signal

```
//*****************************************************************
// Ce programme permet de lire un fichier de donnees et de le scinder en plusieurs fichier.
// Dans ce fichier on divise le signal en 4 signaux de 32 intervales ultrasonore de x usec.
// Ce programme tient compte du nombre d'echantillons recueilli par ligne de balayage du programme
//        Extraction.v (soit Temps_Fin - Temps_Depart = Sample_Fichier)
//
//*****************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
`define Ligne_Fichier 128      // Nombre de lignes desirees par fichier
`define Deep_Varie 643         // Profondeur en terme d'echantillons desirees
`define Sample_Fichier 650     // Nombre d'echantillons par ligne de scan

module Cut_Extract(Clk);

        input Clk;

        reg Cond_Ouvert_R;                          // resultat ouverture du fichier lecture
        reg Cond_Ouvert_W1, Cond_Ouvert_W2;         // resultat ouverture des fichier ecriture
        reg Cond_Ouvert_W3, Cond_Ouvert_W4;

        integer Filer;                              // variable fichier lecture
        integer Filew1, Filew2, Filew3, Filew4;     // variable fichier ecriture
        integer C;                                  // variable d'ecriture de fichier
        integer Division;
        integer T_Division;
        integer Nb_Ligne;
        integer T_Nb_Ligne;
        integer Nb_Sample;
        integer T_Nb_Sample;

        real Decimal;                               // registre de la lecture du fichier
```

```
initial begin

// Ouverture du fichier a lire
                Filer = $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/echo_traite.txt", "r");
                if (Filer == `NULL)
                        Cond_Ouvert_R = 0;
                else
                        Cond_Ouvert_R = 1;
                $display("Code d'ouverture de fichier lecture :", Cond_Ouvert_R);
// fichier1

                Filew1                                                                          =
$fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne.txt", "w");
                if (Filew1 == `NULL)
                        Cond_Ouvert_W1 = 0;
                else
                        Cond_Ouvert_W1 = 1;
                $display("Code d'ouverture de fichier lecture1 :", Cond_Ouvert_W1);

// Initialisation des variables

  C = $fscanf(Filer, " %f \n ", Decimal);
  Division = 0;
  T_Division = 1;
  Nb_Ligne = 1;
  T_Nb_Ligne = 1;
  Nb_Sample = 0;
  T_Nb_Sample = 0;

end
                                // Process de lecture et d'ecriture
always @(negedge Clk) begin

                if( C != `EOF)
                  begin

                        if(Division == 1 & Nb_Sample <= `Deep_Varie)
                        begin
                                $fdisplayb(Filew1, Decimal);
                         end

                        if(Division == 2)
                        begin
$fclose("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne.txt");
                                        Filew2                                          =
$fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne2.txt", "w");
                        if (Filew2 == `NULL)
                                Cond_Ouvert_W2 = 0;
                        else
                                Cond_Ouvert_W2 = 1;

                        $display("Code d'ouverture de fichier lecture2 :", Cond_Ouvert_W2);
```

```
                    T_Division = 3;
                end


                if(Division == 3 & Nb_Sample <= `Deep_Varie)
                 begin
                    $fdisplayb(Filew2, Decimal);
                 end

if(Division == 4)
                begin
$fclose("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne2.txt");
Filew3  =  $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne3.txt",
"w");
                if (Filew3 == `NULL)
                    Cond_Ouvert_W3 = 0;
                else
                    Cond_Ouvert_W3 = 1;
                $display("Code d'ouverture de fichier lecture3 :", Cond_Ouvert_W3);
                T_Division = 5;
                 end

                if(Division == 5 & Nb_Sample <= `Deep_Varie)
                 begin
                    $fdisplayb(Filew3, Decimal);
                 end

                if(Division == 6)
                 begin
$fclose("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne3.txt");
Filew4  =  $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne4.txt",
"w");
                if (Filew4 == `NULL)
                    Cond_Ouvert_W4 = 0;
                else
                    Cond_Ouvert_W4 = 1;
                $display("Code d'ouverture de fichier lecture4 :", Cond_Ouvert_W4);
                            T_Division = 7;
                end

                if(Division == 7 & Nb_Sample <= `Deep_Varie)
                 begin
                    $fdisplayb(Filew4, Decimal);
                end

                if(Division == 8)

        $fclose("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/echo_sectionne4.txt");


            if (Nb_Sample == `Sample_Fichier)
            begin
              Nb_Ligne = T_Nb_Ligne + 1;
```

```
                T_Nb_Sample = 0;
                $display(" Nb_Ligne :", T_Nb_Ligne);
                $display(" Nb_Sample :", Nb_Sample);
              end
          else
              begin
                T_Nb_Sample = Nb_Sample;
                Nb_Ligne = T_Nb_Ligne;
              end

          Division = T_Division;
               end
end

                                        // Process d'identification d'un pick
always @(posedge Clk) begin

      if( C != `EOF)
        begin

          C = $fscanf(Filer, " %f \n ", Decimal);
          Nb_Sample = T_Nb_Sample + 1;
  //$display(" Division :", Division);

          if (Nb_Ligne == `Ligne_Fichier + 1)
            begin
              Division = T_Division + 1;
              T_Nb_Ligne = 1;
            end
          else
            begin
              T_Nb_Ligne = Nb_Ligne;
              Division = T_Division;
                   end
          end
 end
endmodule
```

Décimation

```
// ************************************************************************
// Ce programme permet de lire un fichier de donnees et de le
// decimer selon un facteur de x donnees.
//
// Nicolas V.Veilleux 2005
//
//************************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
```

```
`define Gain 1
`define Decimation 10

module Decimation(clk);

        input clk;

        reg cond_ouvert_r;              // resultat ouverture du fichier lecture
        reg cond_ouvert_w;              // resultat ouverture du fichier ecriture

        integer filer;                  // variable fichier lecture
        integer filew;                  // variable fichier ecriture
        integer c;                      // variable d'ecriture de fichier

        real decimal;                   // registre de la lecture du fichier
        real Tampon;

        integer DataOut;
        integer compteur_mot;
        integer tampon_compteur_mot;
        integer compt;
        integer tamp_compt;

initial begin

// Ouverture du fichier a lire
        filer   =   $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/numeric_in.txt",
"r");
        if (filer == `NULL)
                cond_ouvert_r = 0;
        else
                cond_ouvert_r = 1;
        $display("Code d'ouverture de fichier lecture :", cond_ouvert_r);

// Ouverture du fichier a ecrire
        filew                                                                                   =
$fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/conv_num/numeric_out.txt", "w");
        if (filew == `NULL)
                cond_ouvert_w = 0;
        else
                cond_ouvert_w = 1;
        $display("Code d'ouverture de fichier ecriture :", cond_ouvert_w);

// Premiere lecture de fichier
        c = 0;//$fscanf(filer, " %f \n ", decimal);

// initialisation des variables du convertisseur

        DataOut = 0;
        compteur_mot = 0;
        tampon_compteur_mot = 0;
        compt = 0;
        tamp_compt = 0;
```

```
end

// Process de lecture et d'ecriture
always @(negedge clk) begin

        if( c != `EOF)
        begin
                c = $fscanf(filer, " %f \n ", decimal);
        // Lecture

                if (tampon_compteur_mot == 0)
                $fdisplayb(filew," %d ", DataOut);                    // Ecriture

                compteur_mot = tampon_compteur_mot + 1;
                compt = tamp_compt + 1;
                //$display(" nombre de donnes : ", compt);
        end
end

// Process de conversion de donnees

always @(posedge clk) begin

        if ((decimal*`Gain) <= 0)
                DataOut = 1;
        else
                DataOut =(decimal*`Gain);

        if (compteur_mot == `Decimation)
        tampon_compteur_mot = 0;
        else
        tampon_compteur_mot = compteur_mot;

        tamp_compt = compt;
end
endmodule
```

Convertisseur AD 11 bits

```
//*********************************************************************
// Ce programme permet de lire un fichier de donnees decimal et
// de les convertirs en donnees numerique de N bits
//*********************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
`define NbBits 11                       // Nombre de bits utiliser
`define Range 1.8                       // range de tension positive du signal
`define Offset 0.9                      // Offset qu'on ajoute au signal
`define Gain 1
```

```
module conv_AD_11(clk, vout);

        output [`NbBits-1:0] vout;
        input clk;

        wire [`NbBits-1:0] vout;

        reg cond_ouvert_r;                      // resultat ouverture du fichier lecture
        reg cond_ouvert_w;                      // resultat ouverture du fichier ecriture
        reg [`NbBits-1:0] add;                  // valeur de 1 binaire
        reg [`NbBits-1:0] DataOut;      // Emmagasine la valeur binaire de sortie du converter
        reg [`NbBits-1:0] Tampon;

        integer filer;                  // variable fichier lecture
        integer filew;                  // variable fichier ecriture
        integer c;                      // variable d'ecriture de fichier
        integer exposant;               // variable de calcul exponentiel
        integer i;                      // variable boucle for
        integer EntierBin;

        real decimal;                   // registre de la lecture du fichier
        real Pas;                       // Pas d'amplitude
        real EntierBinDec;

initial begin

// Ouverture du fichier a lire
        filer   =   $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/numeric_in.txt",
"r");
        if (filer == `NULL)
                cond_ouvert_r = 0;
        else
                cond_ouvert_r = 1;
        $display("Code d'ouverture de fichier lecture :", cond_ouvert_r);

// Ouverture du fichier a ecrire
        filew = $fopen("/home/veilleux/DonneeTest/resultat_FEND/fichierdetest_1.txt", "w");
        if (filew == `NULL)
                cond_ouvert_w = 0;
        else
                cond_ouvert_w = 1;
        $display("Code d'ouverture de fichier ecriture :", cond_ouvert_w);

// Premiere lecture de fichier
        c = $fscanf(filer, " %f \n ", decimal);

// initialisation des variables du convertisseur
        add = `NbBits'b00000000001;             // A MODIFIER SI ON MODIFIE NbBits
        DataOut = `NbBits'b00000000000;         // A MODIFIER SI ON MODIFIE NbBits
        Tampon = `NbBits'b00000000000;
        exposant = 1;
```

```
        for(i = 0; i < `NbBits; i = i + 1)       // calcul de l'exposant en base 2
                exposant = exposant * 2;

        Pas = `Range/exposant;
end

// Process de lecture et d'ecriture
always @(negedge clk) begin

        if( c != `EOF)
        begin
                c = $fscanf(filer, " %f \n ", decimal);
        // Lecture
                //$display("allo   : ", decimal);
                $fdisplayb(filew, DataOut);          // Ecriture

        Tampon = `NbBits'b00000000000;               // A MODIFIER SI ON MODIFIE NbBits
        end
end

// Process de conversion de donnees
always @(posedge clk) begin

        if(c != `EOF)
        begin
        EntierBinDec = (((decimal + `Offset)*`Gain)/Pas);
        //$display("valeur de decimal :", decimal);
        if(EntierBinDec > exposant)
        EntierBin = exposant;
        else
        EntierBin = EntierBinDec;
        //$display("valeur de EntierBinDec :", EntierBinDec);

        //$display("valeur de EntierBin :", EntierBin);

        for( i = 0; i < EntierBin ; i = i + 1)
        begin
                Tampon = Tampon + add;
        end

        DataOut = Tampon;
        //$display("valeur de DataOut :", DataOut);
        end
end

assign vout = DataOut;

endmodule
```

Enveloppe IQ

```
//*********************************************************************
// Ce programme permet de filtrer un signal
// Type butterworth d'ordre 4 a 2 sections a modulation IQ
//
// Nom: Nicolas V.Veilleux
// Date: 2005
//
//*********************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
`define Offset 0
`define Gain 1

module Envelop_IQ(Clk);

        input Clk;

         integer Filer;              // Ouverture du fichier comptenant le signal du debut du balayage
        integer Cond_Ouvert_r;              // Condition d'ouverture du fichier lecture
        integer C;
        integer Filew;                      // Ouverture de fichier d'ecriture
        integer Cond_Ouvert_w;              // Condition d'ouverture du fichier d'ecriture
        integer i;
        integer j;
        integer compteur;
        integer T_compteur;

        real IN;
        real VOUT;
        real IN1;

  real M[1:42];
  real REG[1:16];
  real S[1:4];
  real a2[1:4];
  real a3[1:4];
  real b2[1:4];
  real lookup[1:11];

 initial begin

 // Ouverture des differents fichier a lire
            Filer = $fopen("/home/veilleux/DonneeTest/DATA/filtre_in.txt", "r");
    if (Filer == `NULL)
            Cond_Ouvert_r = 0;
    else
            Cond_Ouvert_r = 1;
    $display("Ouverture Start :", Cond_Ouvert_r);

 // Ouverture du fichier d'ecriture
```

```
        Filew = $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/filtre_out.txt", "w");
        if (Filew == `NULL)
                Cond_Ouvert_w = 0;
        else
                Cond_Ouvert_w = 1;

        $display(" Ouverture Echo Traite :", Cond_Ouvert_w);

// Initialisation des variables

        IN = 0;
        VOUT = 0;
        IN1 = 0;

        compteur = 1;
        T_compteur = 1;

        for(i=1; i<=41; i=i+1)
          M[i] = 0;

        for(i=1; i<=16; i=i+1)
        REG[i] = 0;

        S[1] = 0.078;
        S[2] = 0.062;
        S[3] = 1;
        S[4] = 1;

        a2[1] = -1.321;
        a2[2] = -1.049;
        a2[3] = 0;
        a2[4] = 0;

        a3[1] = 0.633;
        a3[2] = 0.296;
        a3[4] = 0;
        a3[3] = 0;

        b2[1] = 2;
        b2[2] = 2;
        b2[3] = 0;
        b2[4] = 0;

        lookup[1] = 1.4142;
        lookup[2] = 0.3129;
        lookup[3] = -0.9080;
        lookup[4] = -1.7820;
        lookup[5] = -1.9754;
        lookup[6] = -1.4142;
        lookup[7] = -0.3129;
        lookup[8] = 0.9080;
        lookup[9] = 1.7820;
```

```
        lookup[10] = 1.9754;
        lookup[11] = 1.4142;

  C = $fscanf(Filer, " %f \n ", IN1);
  IN = (IN1 - `Offset)*`Gain;

end

always @(posedge Clk) begin

        if(C != `EOF)
         begin

 //lookup table pour le dephasage de 45 degres
          M[21] = IN * lookup[compteur];
          M[42] = IN * lookup[12 - compteur];

// Lecture des fichiers
                          j=0;
           for(i=0; i<=21; i=i+21)
            begin
                    M[1+i] = M[21+i] * S[1];
                    M[2+i] = M[1+i] - (REG[1+j] * a2[1]);
                    M[3+i] = M[2+i] - (REG[2+j] * a3[1]);
                    M[4+i] = M[3+i] + (REG[1+j] * b2[1]);
                    M[5+i] = M[4+i] + REG[2+j];

                    M[6+i] = M[5+i] * S[2];
                    M[7+i] = M[6+i] - (REG[3+j] * a2[2]);
                    M[8+i] = M[7+i] - (REG[4+j] * a3[2]);
                    M[9+i] = M[8+i] + (REG[3+j] * b2[2]);
                    M[10+i] = M[9+i] + REG[4+j];

                    M[11+i] = M[10+i] * S[3];
                    M[12+i] = M[11+i] - (REG[5+j] * a2[3]);
                    M[13+i] = M[12+i] - (REG[6+j] * a3[3]);
                    M[14+i] = M[13+i] + (REG[5+j] * b2[3]);
                    M[15+i] = M[14+i] + REG[6+j];

                    M[16+i] = M[15+i] * S[4];
                    M[17+i] = M[16+i] - (REG[7+j] * a2[4]);
                    M[18+i] = M[17+i] - (REG[8+j] * a3[4]);
                    M[19+i] = M[18+i] + (REG[7+j] * b2[4]);
                    M[20+i] = M[19+i] + REG[8+j];

                    j=j+8;
      end

      VOUT = (M[20]*M[20]) + (M[41]*M[41]);
      compteur = T_compteur + 1;
            end
          end
```

```
always @(negedge Clk) begin

        if(C != `EOF)
         begin

// Delais
          C = $fscanf(Filer, " %f \n ", IN1);
          IN = (IN1 - `Offset)*`Gain;

          j=0;
          for(i=0; i<=21; i=i+21)
           begin

            REG[2+j] = REG[1+j];
            REG[4+j] = REG[3+j];
            REG[1+j] = M[3+i];
            REG[3+j] = M[8+i];

            j=j+8;
           end

          $fdisplayb(Filew, VOUT);

          if(compteur > 10)
            T_compteur = 1;
          else
            T_compteur = compteur;
         end
  end
endmodule
```

Enveloppe simple

```
//*********************************************************************
// Ce programme permet de filtrer un signal
// Type butterworth d'ordre 4 a 2 sections
//
// Nom: Nicolas V.Veilleux
// Date: 2005
//
//*********************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
`define Offset 0
`define gain 1

module Envelop_simple(Clk);

        input Clk;
```

```
        integer Filer;              // Ouverture du fichier comptenant le signal du debut du balayage
        integer Cond_Ouvert_r;      // Condition d'ouverture du fichier lecture
        integer C;
        integer Filew;              // Ouverture de fichier d'ecriture
        integer Cond_Ouvert_w;      // Condition d'ouverture du fichier d'ecriture
        integer i;

        real IN;
        real IN1;
        real VOUT;
        real VOUT2;

 real MO[1:9];
 real REGO[1:4];
 real SO[1:2];
 real a2O[1:2];
 real a3O[1:2];
 real b2O[1:2];

initial begin

// Ouverture des differents fichier a lire

        Filer = $fopen("/home/veilleux/DonneeTest/DATA/filtre_in.txt", "r");
   if (Filer == `NULL)
        Cond_Ouvert_r = 0;
   else
        Cond_Ouvert_r = 1;
   $display("Ouverture Start :", Cond_Ouvert_r);

// Ouverture du fichier d'ecriture

        Filew = $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/filtre_out.txt", "w");
        if (Filew == `NULL)
                Cond_Ouvert_w = 0;
        else
                Cond_Ouvert_w = 1;

        $display(" Ouverture Echo Traite :", Cond_Ouvert_w);

// Initialisation des variables

        IN = 0;
        IN1 = 0;
        VOUT2 = 0;

        for(i=1; i<=9; i=i+1)
          MO[i] = 0;

        for(i=1; i<=4; i=i+1)
        REGO[i] = 0;
```

```
        SO[1] = 0.078;
        SO[2] = 0.062;
        a2O[1] = -1.321;
        a2O[2] = -1.049;
        a3O[1] = 0.633;
        a3O[2] = 0.296;
        b2O[1] = 2;
        b2O[2] = 2;

  C = $fscanf(Filer, " %f \n ", IN1);
  IN = IN1 - `Offset;
        VOUT = IN*IN;
end
always @(posedge Clk) begin

        if(C != `EOF)
         begin


// Lecture des fichiers FILTRE

                VOUT = IN * IN;
                MO[1] = VOUT * SO[1];
                MO[2] = MO[1] - (REGO[1] * a2O[1]);
                MO[3] = MO[2] - (REGO[2] * a3O[1]);
                MO[4] = MO[3] + (REGO[1] * b2O[1]);
                MO[5] = MO[4] + REGO[2];
                MO[6] = MO[5] * SO[2];
                MO[7] = MO[6] - (REGO[3] * a2O[2]);
                MO[8] = MO[7] - (REGO[4] * a3O[2]);
                MO[9] = MO[8] + (REGO[3] * b2O[2]);
                VOUT2 = MO[9] + REGO[4];
         end
        end

always @(negedge Clk) begin

        if(C != `EOF)
         begin

// Delais

          C = $fscanf(Filer, " %f \n ", IN1);

          REGO[2] = REGO[1];
          REGO[4] = REGO[3];
          REGO[1] = MO[3];
          REGO[3] = MO[8];

          $fdisplayb(Filew, (VOUT2*`gain));

          IN = IN1 - `Offset;
         end
```

```
  end
endmodule
```

Compression

```
//*********************************************************************
// Ce programme permet de lire un fichier de donnees decimal et de les
// convertirs en donnees numerique de N bits
//
// Nicolas V.Veilleux 2005
//
//*********************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
`define NbDivision 1024
`define Offset 1                        // Valeur la plus eleve du signal
`define Offset2 40                      // Offset qu'on ajoute au signal
`define Samp_ligne 6500                 // Nombre d'echantillon pour une ligne de scan
`define DC 0.0005                       // definition du seuil de coupure de bruit
`define Gain 50

module compresse(clk);

        input clk;

        reg cond_ouvert_r;              // resultat ouverture du fichier lecture
        reg cond_ouvert_w;              // resultat ouverture du fichier ecriture

        integer filer;                  // variable fichier lecture
        integer filew;                  // variable fichier ecriture
        integer c;                      // variable d'ecriture de fichier

        real decimal;                   // registre de la lecture du fichier
        real FacteurDeConvertion;       // Facteur par lequel on convertit la tension d'entree
        real Tampon;

        integer DataOut;
        integer compteur_mot;
        integer tampon_compteur_mot;
        integer compt;
        integer tamp_compt;

initial begin

// Ouverture du fichier a lire
        filer = $fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/numeric_in.txt",
"r");
        if (filer == `NULL)
                cond_ouvert_r = 0;
```

```
        else
                cond_ouvert_r = 1;
        $display("Code d'ouverture de fichier lecture :", cond_ouvert_r);

// Ouverture du fichier a ecrire
        filew                                                                                    =
$fopen("/home/veilleux/DonneeTest/DATA/Secance_50905/Separation/conv_num/numeric_out.txt", "w");
        if (filew == `NULL)
                cond_ouvert_w = 0;
        else
                cond_ouvert_w = 1;
        $display("Code d'ouverture de fichier ecriture :", cond_ouvert_w);

// Premiere lecture de fichier
        c = 0;//$fscanf(filer, " %f \n ", decimal);


// initialisation des variables du convertisseur

        FacteurDeConvertion = `Offset2/`Offset;
        DataOut = 0;
        compteur_mot = 0;
        tampon_compteur_mot = 0;
        compt = 0;
        tamp_compt = 0;
end

// Process de lecture et d'ecriture
always @(negedge clk) begin

        if( c != `EOF)
        begin
                c = $fscanf(filer, " %f \n ", decimal);                  // Lecture

                $fdisplayb(filew," %d ", DataOut);                       // Ecriture

                compteur_mot = tampon_compteur_mot + 1;
                compt = tamp_compt + 1;
                $display(" nombre de donnes : ", compt);
        end
end

// Process de conversion de donnees

always @(posedge clk) begin

        if (((decimal*`Gain) - `DC) <= 0)
                DataOut = 1;
        else
                DataOut =(decimal*`Gain) * FacteurDeConvertion;

        if (compteur_mot == `Samp_ligne)
        tampon_compteur_mot = 0;
```

```
        else
        tampon_compteur_mot = compteur_mot;

        tamp_compt = compt;
end
endmodule
```

CADENCE

Fichier de lecture dans CADENCE

```
//*****************************************************************
// Verilog HDL for "Outils_v_digital", "openread" "functional"
//  Permet d'introduire les données d'un fichier dans l'environnement de cadence
//
//*****************************************************************

`define NULL 0
`define EOF 32'hFFFF_FFFF
`define NbBit 16
`define NbWord 1000000

module openread_163(x);

        output [`NbBit-1:0] x;

        reg clk;
        reg [`NbBit-1:0] memry [0:`NbWord];
        reg [`NbBit:0] Suite;

        wire [`NbBit-1:0] memwrd;
        wire [`NbBit-1:0] x;

        integer i, j, g;

initial begin

        clk = 1'b1;
        g = 0;
end

always begin

        #10 clk = ~clk;
end

initial begin

        $readmemb("/home/veilleux/DonneeTest/resultat_FEND/fichierdetest163.txt", memry);
end

always @(posedge clk) begin

        g = g + 1;
        Suite = memry[g];
end
        assign x = Suite;

endmodule
```

Modèle comportemental du TLA

```
//************************************************************************
// Verilog HDL for "Outils_v_digital", "openread" "functional"
//
// Modele comportemental du TLA
//
//************************************************************************

`include "constants.vams"
`include "disciplines.vams"

module TLA_MOD_1(in, out);

      input in;
      output out;
      electrical in, out;

      parameter real GainA = 6.76; // gain de chaque etage
      parameter real Vol = 0.24; // tension limite d'ajout
      parameter real n = 6;

      real s1;
      real s2;
      real s3;
      real in1;
      real out1;

 analog begin

      if(V(in) < 0.00000013 & V(in) > -0.00000013)
            in1 = 0.000000130;
      else
            in1 = V(in);

      s1 = 1/GainA;
      s3 = log(GainA + 1);

      if(in1 < 0)
       begin
            s2 = log(GainA * (-in1) / Vol);
            out1 = -((n + s1 + (s2 / s3)) * Vol);
       end

      if(in1 > 0)
       begin
            s2 = log(GainA * in1 / Vol);
            out1 = ((n + s1 + (s2 / s3)) * Vol);
       end

     V(out) <+ out1;
 end
 endmodule
```

Modèle comportemental du TGC

```
//*********************************************************************
// Verilog HDL for "Outils_v_digital", "openread" "functional"
//
//  Model comportemental du TGC
//
//*********************************************************************

`include "constants.vams"
`include "disciplines.vams"

module TGC_Prog_1(clk, profondeur, in, out);

      input clk, profondeur, in;
      output out;

      electrical clk, profondeur, in, out;

  // En dB
      parameter real GInit = 1;
      parameter real GProche = 2;
      parameter real GMoyen = 6;
      parameter real GLoin = 1;

  // Pour la transition
      parameter real delais = 1p;
      parameter real fall = 1p;
      parameter real rise = 1p;

  // En Volt
      parameter real Vth = 0.9;

  // En V/V
      real Init;
      real Proche;
      real Moyen;
      real Loin;

      real out1;
      real in1;
      real compteur;

 analog begin

      @(initial_step("ac", "dc", "tran", "xf"))
       begin
            compteur = 0;
            out1 = 0;
            in1 = 0;
            Init = pow(10, GInit/20);
            Proche = 0;
            Moyen = 0;
```

```
        Loin = 0;
    end

    in1 = V(in) * Init;

    if(compteur <= 8)
     begin
        Proche = pow(10, GProche/20);
        out1 =  in1 * pow(Proche,compteur);
     end

    if(compteur > 8 & compteur <= 20)
     begin
        Moyen = pow(10, (50 + (GMoyen * (compteur - 8)))/20);
        out1 =  in1 * Moyen;
     end

    if(compteur > 20)
     begin
        Loin = pow(10, (50 + (GLoin * (compteur - 20)))/20);
        out1 =  in1 * Loin;
     end

     @(cross(V(clk)-Vth, 1))
     begin
        compteur = compteur + 1;
     end

     if(compteur > V(profondeur))
        compteur = 1;

  V(out) <+ out1;      // transition(out1, delais, fall, rise);
 end
endmodule
```

Fichier d'écriture dans CADENCE

```
//*********************************************************************
// Verilog HDL for "Outils_v_digital", "openread" "functional"
//
//  Permet de faire l'acquisition de données dans cadence
//
//*********************************************************************

`include "constants.vams"
`include "disciplines.vams"

module openwrite3(clk, in);

      input in;
```

```
    input clk;
    electrical clk, in;

    parameter real Vth = 0.9;

    integer filer;

analog  begin

    @(initial_step)
     begin
          filer = 0;
          filer = $fopen("/home/veilleux/DonneeTest/resultat_FEND/file_adc3.txt");
          if(filer == 0)
               $strobe(" Pas ouvert ");
          else
               $strobe(" Ouvert ");
     end

    @(cross(V(clk)-Vth, 1))
     begin
          $fdisplay(filer,"%g", V(in));
     end

    @(final_step)
     begin
          $fclose(filer);
     end
 end
endmodule
```

Banc d'essai du TGC (contrôleur et décodeur)

```
//*********************************************************************
// Verilog HDL for "TGC_layout", "controleur" "functionnal"
//
// Controleur et decodeur pour le TGC
//
// Nicolas V.Veilleux
//
//*********************************************************************

`define NbLigne 160

module controleur(Clk,out);

          input Clk;
          output [`NbLigne:1] out;

          wire [`NbLigne:1] out;
```

```
        reg [`NbLigne:1] Sortie;

        integer Initiale;
        integer Proche;
        integer Moyen;
        integer Loin;
        integer compteur;
        integer tamp_compteur;
        integer i,j,f;
        integer Somme;

initial begin

  for (i=1; i <= `NbLigne/2; i=i+1)
     begin
          Sortie[i]=1'b1;
          Sortie[80+i]=1'b0;
     end

  Initiale = 1;
  Proche = 2;
  Moyen = -6;
  Loin = 0;
  compteur = 1;
  tamp_compteur = 0;
  Somme = 0;
end

always@(posedge Clk) begin

if (compteur < 9)
  Somme = Initiale + compteur*Proche;

else if (compteur >= 9 & compteur < 21)
  Somme = 50 + (compteur - 8)*Moyen ;

else if (compteur >= 21 & compteur < 26)
  Somme = 50 + (compteur - 20)*Loin;

else
  Somme = 80;

if (compteur == 20)
  tamp_compteur = 0;
else
  tamp_compteur = compteur;

for(j=1; j<=Somme; j=j+1)
     begin
          Sortie[j]=1'b0;
          Sortie[80+j]=1'b1;
     end
```

```
for(f=Somme+1; f<=`NbLigne/2; f=f+1)
      begin
            Sortie[f]=1'b1;
            Sortie[80+f]=1'b0;
      end

if (Somme >40)
      Sortie[101]=1'b0;

if (Somme >60)
      Sortie[121]=1'b0;

end

always@(negedge Clk) begin

  compteur = tamp_compteur + 1;

end

assign out = Sortie;
endmodule
```

MATLAB

Convertisseur de balayage

```
%-------------------------------------------------------------------------------------------------
%Programme de conversion des coordonnees polaires a cartesienne pour une
% profondeur de 10 cm.
%
%Nicolas V.Veilleux 2005
%-------------------------------------------------------------------------------------------------

%initialisation des variables

Profondeur = 10;                        %cm
Nombre_Echantillon = 6500;              %Nb echantillon par ligne

Hauteur_Max = 10;
Largeur_Max = 7;
Pixel_Max_L = 320;
Pixel_Max_H = 240;                      %La limitation est cause par la hauteur

teta = 0;
ligne = 0;
grandeurF = 0;
limite =0;

teta = 0.7*pi/180;                      %augmentation de l'angle de 0.7 degre
limite = 44.8*pi/180;                   %L'angle le plus grand
R = Profondeur/Nombre_Echantillon;      %position de l'echantillon
Rmodif = 3;                             %Decentrer le point de depart des lignes de scan

Active_NNI = 0;                         %Mettre a 1 pour activer NNI a droite (256 lignes)
Active_LIN = 0;                         %Mettre a 1 pour activer LIN (256 lignes)
Active_NORMAL = 0;                      %Mettre a 1 pour avoir ligne unique (128 lignes)
Active_NNI_LIN = 1;                     %Mettre a 1 pour avoir mixte NNI et LIN (512 lignes)

NNI_LIN_DATA = 126*Nombre_Echantillon;
MIXTE_DATA = 384*Nombre_Echantillon;

Facteur_H = 0.0442; %  = Largeur_Max/Pixel_Max_L;     %facteur de conversion en hauteur

% Ouverture du fichier
load ('E:\Projet Ultrason\Simulation\matlab_ultrason\Resultat\data_num_10.txt');
Data_fichier = data_num_10;
grandeurF = size(data_num_10,1) + (Active_NNI*NNI_LIN_DATA) + (Active_LIN*NNI_LIN_DATA) +
(Active_NNI_LIN*MIXTE_DATA);

ima(1:grandeurF,1:4) = 0;               % La sortie ce presente sous forme de 4
                                        % colonnes (compteur, X, Y, Pixels)

%-----------------------------------------------------NORMAL-----------------------------------
if Active_NORMAL == 1
```

```
for jj1 = 0:63          %Ligne de balayage
  for ii1 = 1:Nombre_Echantillon

     X = int2str((160 - (((ii1*R)+Rmodif)*sin(limite - (ligne*teta))/Facteur_H)+ 0.5));
     Y = int2str((((ii1*R)+Rmodif)*cos(limite - (ligne*teta))/Facteur_H)+ 0.5);
     ima(ii1 + (Nombre_Echantillon*ligne),2) = str2num(X);
     ima(ii1 + (Nombre_Echantillon*ligne),3) = str2num(Y);
     ima(ii1 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii1 + (Nombre_Echantillon*ligne),1);
     ima(ii1 + (Nombre_Echantillon*ligne),1) = (ii1 + (Nombre_Echantillon*ligne));

  end
  ligne = ligne + 1;
end

if ligne >=64
  for jj2 = 0:62
    for ii2 = 1:Nombre_Echantillon

       X = int2str(160 + (((ii2*R)+Rmodif)*sin((jj2*teta))/Facteur_H)+ 0.5);
       Y = int2str((((ii2*R)+Rmodif)*cos((jj2*teta))/Facteur_H)+ 0.5);
       ima(ii2 + (Nombre_Echantillon*ligne),2) = str2num(X);
       ima(ii2 + (Nombre_Echantillon*ligne),3) = str2num(Y);
       ima(ii2 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii2 + (Nombre_Echantillon*ligne),1);
       ima(ii2 + (Nombre_Echantillon*ligne),1) = (ii2 + (Nombre_Echantillon*ligne));
    end
    ligne = ligne + 1;
  end
end
dlmwrite('E:\Projet Ultrason\Simulation\matlab_ultrason\Resultat\Data_10.txt', ima, '\t',1,1);
end
% Les facteurs se trouvent en divisant la largeur maximal de la position de
% l'echantillon (pour une demi zone de balayage) par le nombre de pixel sur
% cette largeur. Il en est de meme avec la hauteur.

%-------------------------------------------------------------NNI-------------------------------------

if Active_NNI == 1
  for jj1 = 0:128          %Ligne de balayage
    for ii1 = 1:Nombre_Echantillon

       X = int2str((160 - (((ii1*R)+Rmodif)*sin(limite - (ligne*teta/2))/Facteur_H)+ 0.5));
       Y = int2str((((ii1*R)+Rmodif)*cos(limite - (ligne*teta/2))/Facteur_H)+ 0.5);
       ima(ii1 + (Nombre_Echantillon*ligne),2) = str2num(X);
       ima(ii1 + (Nombre_Echantillon*ligne),3) = str2num(Y);

       if mod((jj1 + 1),2) == 0

         ima(ii1 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii1 + (Nombre_Echantillon*(ligne-
1)/2),1);

       else
         ima(ii1 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii1 + (Nombre_Echantillon*ligne/2),1);
       end
```

```
        ima(ii1 + (Nombre_Echantillon*ligne),1) = (ii1 + (Nombre_Echantillon*ligne));

    end
    ligne = ligne + 1;
end

if ligne >=129
    for jj2 = 1:124
        for ii2 = 1:Nombre_Echantillon

            X = int2str(160 + (((ii2*R)+Rmodif)*sin((jj2*teta/2))/Facteur_H)+ 0.5);
            Y = int2str((((ii2*R)+Rmodif)*cos((jj2*teta/2))/Facteur_H)+ 0.5);
            ima(ii2 + (Nombre_Echantillon*ligne),2) = str2num(X);
            ima(ii2 + (Nombre_Echantillon*ligne),3) = str2num(Y);

            if mod((jj2 + 1),2) == 0

                ima(ii2 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii2 + (Nombre_Echantillon*(ligne-
1)/2),1);
            else
                ima(ii2 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii2 + (Nombre_Echantillon*ligne/2),1);
            end

            ima(ii2 + (Nombre_Echantillon*ligne),1) = (ii2 + (Nombre_Echantillon*ligne));
        end
        ligne = ligne + 1;
    end
end
dlmwrite('E:\Projet Ultrason\Simulation\matlab_ultrason\Resultat\Data_10.txt', ima, '\t',1,1);
end

%-----------------------------------------------------------LIN-----------------------------------------

if Active_LIN == 1
    for jj1 = 0:128          %Ligne de balayage
        for ii1 = 1:Nombre_Echantillon

            X = int2str((160 - (((ii1*R)+Rmodif)*sin(limite - (ligne*teta/2))/Facteur_H)+ 0.5));
            Y = int2str((((ii1*R)+Rmodif)*cos(limite - (ligne*teta/2))/Facteur_H)+ 0.5);
            ima(ii1 + (Nombre_Echantillon*ligne),2) = str2num(X);
            ima(ii1 + (Nombre_Echantillon*ligne),3) = str2num(Y);

            if mod((jj1 + 1),2) == 0

                ima(ii1 + (Nombre_Echantillon*ligne),4) = ((Data_fichier(ii1 + (Nombre_Echantillon*((ligne -
1)/2)),1)) + (Data_fichier(ii1 + (Nombre_Echantillon*((ligne+1)/2)),1)))/2;

            else
                ima(ii1 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii1 + (Nombre_Echantillon*ligne/2),1);
            end

            ima(ii1 + (Nombre_Echantillon*ligne),1) = (ii1 + (Nombre_Echantillon*ligne));
```

```
  end
  ligne = ligne + 1;
  end

if ligne >=129
  for jj2 = 1:124
    for ii2 = 1:Nombre_Echantillon

      X = int2str(160 + (((ii2*R)+Rmodif)*sin((jj2*teta/2))/Facteur_H)+ 0.5);
      Y = int2str((((ii2*R)+Rmodif)*cos((jj2*teta/2))/Facteur_H)+ 0.5);
      ima(ii2 + (Nombre_Echantillon*ligne),2) = str2num(X);
      ima(ii2 + (Nombre_Echantillon*ligne),3) = str2num(Y);

      if mod((jj2 + 1),2) == 0

      ima(ii2 + (Nombre_Echantillon*ligne),4) = ((Data_fichier(ii2 + (Nombre_Echantillon*(ligne-
1)/2),1)) + (Data_fichier(ii2 + (Nombre_Echantillon*((ligne+1)/2)),1)))/2;
      else
      ima(ii2 + (Nombre_Echantillon*ligne),4) = Data_fichier(ii2 + (Nombre_Echantillon*ligne/2),1);
      end

      ima(ii2 + (Nombre_Echantillon*ligne),1) = (ii2 + (Nombre_Echantillon*ligne));
    end
    ligne = ligne + 1;
  end
end
dlmwrite('E:\Projet Ultrason\Simulation\matlab_ultrason\Resultat\Data_10.txt', ima, '\t',1,1);
end

%---------------------------------------------------- LIN + NNI ------------------------------------

if Active_NNI_LIN == 1
  for jj1 = 0:256          %Ligne de balayage
    for ii1 = 1:Nombre_Echantillon

      X = int2str((160 - (((ii1*R)+Rmodif)*sin(limite - (ligne*teta/4))/Facteur_H)+ 0.5));
      Y = int2str((((ii1*R)+Rmodif)*cos(limite - (ligne*teta/4))/Facteur_H)+ 0.5);
      ima(ii1 + (Nombre_Echantillon*ligne),2) = str2num(X);
      ima(ii1 + (Nombre_Echantillon*ligne),3) = str2num(Y);

      if mod((jj1 + 1),4) == 0
        ima(ii1 + (Nombre_Echantillon*ligne),4) = (Data_fichier  (ii1 + (Nombre_Echantillon*(ligne +
1)/4),1));
      elseif mod((jj1 + 1),4) == 1
        ima(ii1 + (Nombre_Echantillon*ligne),4) = (Data_fichier (ii1 +
(Nombre_Echantillon*ligne/4),1));
      elseif mod((jj1 + 1),4) == 2
        ima(ii1 + (Nombre_Echantillon*ligne),4) = (Data_fichier  (ii1 + (Nombre_Echantillon*(ligne -
1)/4),1));
      elseif mod((jj1 + 1),4) == 3
        ima(ii1 + (Nombre_Echantillon*ligne),4) = ((Data_fichier(ii1 + (Nombre_Echantillon*((ligne -
2)/4)),1)) + (Data_fichier(ii1 + (Nombre_Echantillon*((ligne + 2)/4)),1)))/2;
```

```
        end

        ima(ii1 + (Nombre_Echantillon*ligne),1) = (ii1 + (Nombre_Echantillon*ligne));

  end
  ligne = ligne + 1;
end

if ligne >=257
  for jj2 = 1:249
     for ii2 = 1:Nombre_Echantillon

        X = int2str(160 + (((ii2*R)+Rmodif)*sin((jj2*teta/4))/Facteur_H)+ 0.5);
        Y = int2str((((ii2*R)+Rmodif)*cos((jj2*teta/4))/Facteur_H)+ 0.5);
        ima(ii2 + (Nombre_Echantillon*ligne),2) = str2num(X);
        ima(ii2 + (Nombre_Echantillon*ligne),3) = str2num(Y);

        if mod((jj2 + 1),4) == 0
          ima(ii2 + (Nombre_Echantillon*ligne),4) = (Data_fichier  (ii2 + (Nombre_Echantillon*(ligne +
1)/4),1));
        elseif mod((jj2 + 1),4) == 1
          ima(ii2      +      (Nombre_Echantillon*ligne),4)      =      (Data_fichier              (ii2      +
(Nombre_Echantillon*ligne/4),1));
        elseif mod((jj2 + 1),4) == 2
          ima(ii2 + (Nombre_Echantillon*ligne),4) = (Data_fichier  (ii2 + (Nombre_Echantillon*(ligne -
1)/4),1));
        elseif mod((jj2 + 1),4) == 3
          ima(ii2 + (Nombre_Echantillon*ligne),4) = ((Data_fichier(ii2 + (Nombre_Echantillon*((ligne -
2)/4)),1)) + (Data_fichier(ii2 + (Nombre_Echantillon*((ligne + 2)/4)),1)))/2;
        end

        ima(ii2 + (Nombre_Echantillon*ligne),1) = (ii2 + (Nombre_Echantillon*ligne));
     end
     ligne = ligne + 1;
  end
end
dlmwrite('E:\Projet Ultrason\Simulation\matlab_ultrason\Resultat\Data_10.txt', ima, '\t',1,1);
end
```

Affichage

```
%-------------------------------------------------------------------------------------------------------
%Programme d'affichage de données cartésiennes
% profondeur de 10 cm.
%
%Nicolas V.Veilleux 2005
%-------------------------------------------------------------------------------------------------------

ima1(1:240,1:320)= 1;
```

```
load ('E:\Projet Ultrason\Simulation\matlab_ultrason\Resultat\Data_10.txt');
dataout3=Data_10;

 for i = 1:size(dataout3,1)

    if (dataout3(i,2) == 0)
        dataout3(i,2) = 1;
    end

    if (dataout3(i,3) == 0)
        dataout3(i,3) = 1;
    end
 if(dataout3(i,3) <= 300 &  dataout3(i,3) > 60 & dataout3(i,2) <= 320 &  dataout3(i,2) >= 0)
      if (ima1(dataout3(i,3)-60,dataout3(i,2)) <=  dataout3(i,4))
           ima1(dataout3(i,3)-60,dataout3(i,2)) =  dataout3(i,4);
      end
 end
end
figure(610);
h = image(zeros(240,320,3));
colormap(gray(40));
image(ima1);
```