

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE
DEPARTEMENT D'ELECTRONIQUE
Mémoire de fin d'études

EN VUE DE L'OBTENTION DU DIPLOME
D'INGENIEUR D'ETAT

المدرسة الوطنية المتعددة التخصصات
المكتبة
BIBLIOTHEQUE
Ecole Nationale Polytechnique

THEME

**Séparation autodidacte
de sources au second ordre:
Description VHDL et
Implémentation de SOBI**

Etudié et soutenu par :
M. DIOP BARA et
M^{lle} GUERFI KELTOUM

Devant le jury composé de:
M. BerkaniProfesseur (E.N.P).....Président
M. SaadounDocteur d'état (ENP)Examineur
M. Bousbia S.....Maître de Conférence.....Promoteur
M. Bélouchrani A.....Docteur d'Etat(ENP).....Co-promoteur

Promotion 2001

Je dédie ce modeste travail :

*A ma très chère mère, pour
tous ses sacrifices.
A mon père,
A mon regretté grand-père, Qu'ALLAH l'accueille
dans Son Vaste Paradis,
A mon oncle Laye que je ne pourrai oublier,
A mon grand-frère et plus grand ami Moussa .
A mes frères et sœurs bien aimés,
A toute ma famille,
A mes très chers amis Abou, Bamba, Badiane, Samba ...
A mon nouveau frère Raphaël,
A toute la nation algérienne,
A ma binôme et sa famille
A tous mes collègues,
A tous mes compatriotes,
A tous mes amis algériens,
A tous ceux qui oeuvrent pour la Paix,
la Justice, et la fraternité,
Et enfin à ma future épouse.*

Diadieuf.

Je dédie ce modeste travail :

*A mes chers parents,
A ma chère grand-mère,
A mes chers soeurs
A mes chers frères,
A mes tantes et oncles,
A mes cousins et cousines,
A Hakim,
Et à tous mes amis*

Keltoum.

Remerciements

Tout d'abord et avant tout, Al hamdoulilah.

Que Monsieur Bousbia et Monsieur Bélouchrani, nos encadreurs, trouvent ici l'expression de notre profonde gratitude pour nous avoir proposé un sujet de recherche intéressant, actuel et plein d'avenir. Qu'ils sachent que leurs conseils avisés, leurs soutiens et leurs encouragements nous ont été très précieux. Ce fût un immense privilège de travailler sur ce thème et une chance exceptionnelle d'avoir Mr. Bousbia comme promoteur. Nous tenons à le remercier de tout notre cœur pour avoir dirigé ce travail, pour l'aide précieuse qu'il nous a apportée et pour le travail considérable qu'il a consacré au suivi et à l'élaboration de ce mémoire. Ce modeste travail n'est que le reflet de ses conseils, suggestions, et revigorants encouragements.

Nous tenons également à exprimer toute notre reconnaissance à :

M. Berkani, Professeur à l'Ecole Nationale Polytechnique, qui nous a fait l'honneur de présider notre jury,

M. Saadoun, Maître de conférence et Directeur de Recherches à l'Ecole Nationale Polytechnique, de nous avoir honorés en acceptant de juger notre travail.

Nous remercions :

- Monsieur Boussefsaf (Chef du département Hardware à l'E.N.S.I (Entreprise Nationale des Systèmes Informatiques))
- Monsieur Djamel
- Monsieur Saadoun, de ses questions pertinentes et indicatrices,
- M. Sami, et l'ensemble des occupants du Labo. Cette ambiance conviviale et sympathique ainsi que les nombreux échanges d'idées ont été fructueux et ont contribué à cette synergie qui fait que le Labo est plus que la somme individuelle de ses membres.
- Monsieur Oudjida (Département Electronique au C.D.T.A (Centre de Développement de Techniques Avancées)
- Tous nos correspondants du monde entier en particulier Dr P.Ashenden (Dept. Computer Science, University of Adelaide, South Australia) et M. Charles Shelor de VHDL.org.

Pour leur aide précieuse témoin de leur sens désintéressé du partage de connaissances.

Notre vive reconnaissance va également à l'endroit de tous nos professeurs qui n'ont ménagé aucun effort pour nous donner un enseignement de qualité durant toute notre scolarité et qui , en fait sont la base de tout ceci.

Enfin, nous adressons nos remerciements à toutes les personnes qui, de près ou de loin, directement ou indirectement, nous ont soutenu, d'une manière ou d'une autre, pour l'aboutissement de notre travail.

A tous, MERCI.

DIOP & GUERFI

ملخص

يخص هذا البحث دراسة طريقة الفصل المجهول للمنابع من الرتبة الثانية (SOBI) وبرمجتها باللغة الوصفية (VHDL) وتتعلق هذه الطريقة بفصل الإشارات المقاسة للحصول على إشارات متناسبة طرديا مع إشارات المنبع، مع الحرص على أن تكون هذه الأخيرة مستقرة و المزيج لحضي و خطي.

كلمات مفتاحية : الفصل المجهول، المنابع، مصفوفة المزج، الحساس، VHDL.

Abstract

The aim of this job is principally to implement SOBI(Second Order Blind Identification), a particular algorithm of Blind Sources Separation(BSS).

Actually, BSS has received attention because of its potential applications in signal processing such as in speech recognition systems, telecommunications and medical signal processing. The goal of BSS is to recover independent sources given only sensor observations that are unknown linear and instantaneous mixtures of the unobserved independent source signals.

We have shared out this work into two main parts that lie like this:

In a first step, we have given a detailed description of Blind Sources Separation followed by a full study of the basis principles of SOBI.

In a second part, after an introduction to hardware description languages, we subsequently lead down a complete development of our application in VHDL using the fundamental tools of XILINX FPGA. Extensive simulations and implementations have been performed to demonstrate the power of the learning algorithm.

Keys words : blind separation, SOBI, VHDL.

Résumé

Ce projet est consacré à l'étude et à l'implémentation de l'une des méthodes de Séparation Aveugle de Sources. L'objectif de la séparation autodidacte est de retrouver à quelques paramètres près des sources à partir d'un mélange instantané et linéaire, aucune condition a priori sur les procédés du mixage des signaux sources ni même aucune hypothèse sur leurs natures n'étant disponibles.

L'algorithme sollicité dans ce travail est SOBI (Second Order BLIND Identification), méthode d'estimation des signaux sources usant des statistiques du second ordre.

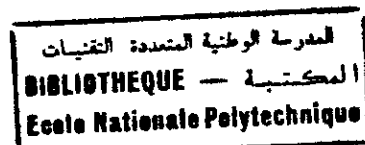
En utilisant le VHDL comme langage de description hardware et le logiciel XILINX, nous avons introduit une complète description de notre application avant de la valider en dernier lieu par une simulation et une implémentation.

Mots clés : séparation aveugle, sources, capteurs, SOBI, VHDL.

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Sommaire

Liste des figures	
Abréviations	
Préambule	
INTRODUCTION GENERALE	1
CHAPITRE I : QU'ENTENDRE PAR SEPARATION AVEUGLE DE SOURCES ?	
I.1. Introduction	3
I.2. Historique.....	3
I.2.1. Les différentes approches.....	3
Les deux écoles	3
I.2.2. Les résolutions.....	4
I.2.2.a. Les algorithmes non-linéaires	4
I.2.2.b. Les algorithmes dans un environnement réel.....	5
I.2.3. Les travaux actuels.....	5
I.3. Position du problème	6
I.3.1 Les méthodes de résolution.....	6
I.3.2. Première définition	7
I.4. Résumé.....	9
I.5. Conclusion.....	9
CHAPITRE II : RAPPELS	
II.1. Introduction.....	10
II.2. Les capteurs	10
II.2.1 Définition	10
II.2.2 Les principaux types de capteurs.....	10
II.2.3 Grandeurs d'influence.....	11
II.3 Les sources	12
II.3.1. Définition.....	12
II.3.2. Les types de sources.....	12
II.3.2.a. Source analogique	12
II.3.2.a. Source discrète	12
II.3.3. Exemples.....	12
II.4. Conclusion.....	13
CHAPITRE III : METHODOLOGIE DE TRAITEMENT	
III.1. Introduction	14
III.2. Les Mélanges instantanés	14
III.3. Notions d'identifiabilité et d'identification	16
III.3.1. L'identifiabilité	16
III.3.2. . L'identification	17
III.3.2. . La séparation	18
III.4. Conclusion	18



CHAPITRE IV : APPLICATIONS DE LA SEPARATION AVEUGLE

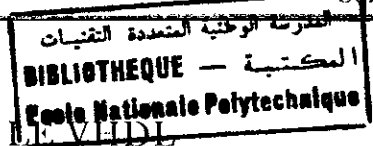
IV.1. Introduction.....	19
IV.2. Les applications.....	19
IV.2.1. Les télécommunications.....	19
IV.2.2. La séparation d'images.....	19
IV.2.3. La reconnaissance de la parole.....	20
IV.2.4. Le traitement du signal sismique.....	20
IV.3. SOBI	20
Exemple d'une application de SOBI	21
IV.3. Conclusion.....	22

CHAPITRE V : SEPARATION AVEUGLE AU SECOND ORDRE : SOBI

V.1. Introduction	23
V.2. Bases de la formulation du problème.....	23
V.3. Principe de base de l'algorithme SOBI.....	23
V.3.1. Les hypothèses	23
V.3.1.a. Les sources.....	24
V.3.1.b. Les perturbations.....	24
V.3.2. Développement	25
V.3.3. La normalisation du problème.....	26
V.3.4. La diagonalisation conjointe	27
V.3.4.a. Principe.....	27
V.3.4.b. Démonstration.....	28
V.3.4.c. Implantation de la diagonalisation conjointe.....	28
V.4. Conclusion.....	30

CHAPITRE VI : INTRODUCTION SUR LE LANGAGE DE DESCRIPTION VHDL

VI.1. Introduction	31
VI.2. A propos du VHDL.....	31
VI.3. Pourquoi le langage de description VHDL?.....	31
VI.4. La conception VHDL.....	32
VI.5. Les niveaux d'abstraction.....	33
VI.5.1. Le niveau système.....	33
VI.5.2. Le niveau algorithmique.....	34
VI.5.3. Le niveau RTL.....	34
VI.5.4. Le niveau logique.....	35
VI.6. Conclusion.....	36



CHAPITRE VII : DEVELOPPEMENT : SOBI ET LE VHDL	
VII.1. Introduction.....	37
VII.2. Estimation du blanchisseur.....	38
VII.2.1 Estimation de la matrice de covariance.....	38
VII.2.2 Diagonalisation.....	38
VII.2.2.a. Détermination des valeurs propres de $\hat{R}(0)$	39
VII.2.2.b. Détermination de vecteurs propres.....	43
VII.2.2.c. Estimation de la puissance du bruit.....	45
VII.2.2.d. La matrice de blanchement \hat{W}	45
VII.3. Blanchiment des données.....	45
VII.4. Estimation des K matrices d'intercovariances $\hat{R}(lk)$ de $\hat{x}(t)$	46
VII.5. Diagonalisation conjointe approchée des matrices d'intercovariances	46
VII.6. Estimation de la matrice de mélange	48
VII.7. Estimation des signaux sources.....	49
VII.8. Conclusion.....	50
CHAPITRE VIII : La technologie FPGA	
VIII.1. Introduction.....	51
VIII.2. Les technologies de conception.....	51
VIII.2.1. Les circuits sur mesure (Full Custom)	51
VIII.2.2. Les circuits précaractérisés.....	51
VIII.2.3. Les circuits prédiffusés.....	51
VIII.2.4. Les composants programmables.....	51
VIII.3. Les FPGA.....	52
VIII.4. La conception avec l'XILINX.....	53
VIII.5. Conclusion.....	55
CHAPITRE IX : LES RESULTATS	
IX.1. Introduction.....	56
IX.2. Le management du projet VHDL.....	56
IX.2.1. L'écriture des conditions.....	57
IX.2.2. Notre "top level"	57
IX.2.3. La Conception Détaillée.....	59
IX.2.4. Exécution.....	59
IX.2.5. Vérification par composants.....	59
IX.2.6. Intégration.....	60
IX.3. Le code de SOBI en VHDL.....	60
IX.3.1. Composition générale d'une description VHDL.....	60
IX.3.2. Les outils de développement.....	62



IX.4. Analyse des résultats.....	..62
IX.4.1. La synthèse.....	..63
IX.4.2. La simulation fonctionnelle64
IX.4.3. L'implémentation.....	..65
IX.5. Les performances.....	..67
IX.6. Conclusion.....	..69

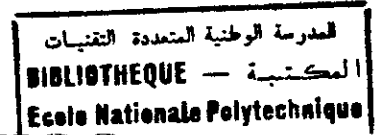
CONCLUSION GENERALE	70
---------------------------	----

ANNEXES

Annexe B: Simulations

Annexe C: Implémentations

REFERENCES BIBLIOGRAPHIQUES



Liste des figures

Chapitre I : Qu'entend-on par séparation aveugle de sources

1-1	Schéma simplifié de la séparation de sources... .. 7
-----	--

Chapitre III : Méthodologie de traitement

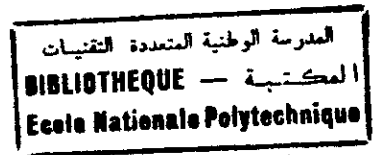
3-1	Modèle de séparation aveugle... .. 14
-----	---------------------------------------

Chapitre VI : Introduction sur le langage de description VHDL

6-1	Le graphe « Y »... .. 34
-----	--------------------------

Chapitre VII : Développement en VHDL

7-1	Algorithme SOBI... .. 37
7-2	Organigramme de Leverrier... .. 40
7-3	Organigramme de calcul de la trace... .. 41
7-4	Organigramme de Graeffe... .. 43
7-5	Organigramme de calcul des vecteurs propres... .. 44
7-6	Organigramme de la diagonalisation conjointe... .. 47
7-7	Organigramme de Greville... .. 49



<i>Chapitre VIII: La technologie FPGA</i>	
8-1	Configuration interne d'un CLB (XC4000)52
8-2	Configuration interne d'un IOB (XC4000)..... 53
8-3	La synthèse avec l'XILINX 54

<i>Chapitre IX : Les Résultats</i>	
9-1	Management du projet.. .. 56
9-2	Conception VHDL61
9-3	Convention d'Xilinx... ..63
9-4	Waveform de l'estimation de R(0)..... 64
9-5	Circuit correspondant à l'estimée de R(0).)..... 65
9-6	Distribution des CLB et des IOB... .. 66

المدرسة الوطنية المتعددة الفنون
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Abréviations

SOBI	Second Order Blind Identification
ISI	Inter Symbol Interference
GSM	Global System for Mobil communication
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
SDMA	Space Division Multiple Access
HOS	Higher Order Statistics
SOS	Second Order Statistics
HMM	Hided Markov Model
LP	Linear Prediction
ML	Maximum likelihood
CAO	Conception assistée par ordinateur
FPGA	Field Programmable Gate Array
PLD	Programmable Logic Device
PROM	Programmable Read Only Memory
PAL	Programmable Array Logic
PLA	Programmable Logic Array

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

VLSI	Very Large Scale of Integration
MPGA	Mask Programmable Gate Array
CLB	Configurables Logic Blocks
IOB	Inputs/Outputs Blocks
LUT	Look-Up Tables
BSS	Blind Source Separation
ULC	Unité Logique Combinatoire

Préambule

Depuis bien longtemps, règne à peu près ce que l'on nomme « une guerre des ondes ». A qui appartiennent les ondes hertziennes ? Aux commerçants, aux militaires, aux scientifiques, ou à l'humanité entière ? La question est devenue urgente depuis l'explosion des télécommunications. Des radiotélescopes détectent des émissions radio en provenance de galaxies particulières, difficiles à détecter sur d'autres longueurs d'ondes. La prolifération des liaisons hertziennes les aveugle.

Le 8 Mai 2000, à Istanbul, s'est tenue une conférence internationale dans laquelle se sont affrontés les responsables des armées, les géants de l'électronique et les radioastronomes.

Au cœur du débat : le système Bluetooth de liaison radio à courte durée entre les micro-ordinateurs et leurs périphériques. Soutenu par des géants de l'électronique tels qu'Ericsson, Nokia, Toshiba, Dell Computer et IBM, Bluetooth exploite un canal radio situé sur la bande des 2.45Gigahertz. Or, l'armée française occupe partiellement celle-ci. Pour les militaires, il est hors de question de céder sans compensations.

Que se passe-t-il du côté des radioastronomes ? Ces derniers écoutent le ciel principalement sur trois groupes de longueurs d'ondes : décimétriques, métriques et décamétriques. Or le premier groupe centré sur une fréquence de 1.4Gigahertz, se trouve déjà encadré par les fréquences qu'exploitent les réseaux de la téléphonie mobile (0.9GHZ à 1.8GHZ). Le second groupe de fréquences concernent essentiellement l'étude du soleil (entre 200MHZ et 600MHZ). Cette fois, ce sont les réseaux de télévisions qui constituent une véritable nuisance. Le dernier groupe, plus vaste et est absolument irremplaçable puisqu'il est actuellement le seul à permettre l'observation des planètes extra-solaires ! Une plage de fréquences pourtant déjà très encombrée par les satellites en orbite basse, les communications militaires.

A qui donnera-t-on la priorité ?

« Avec une séparation à la réception de différentes sources qui sont les leurs et dans les stations différentes, et chacune se chargeant après séparation de recueillir la ou les sources désirées et d'éliminer les autres considérées des lors comme indésirables, chacun se chargeant de coder ses envois à volonté » serait-on presque tenté d'avancer mais seulement, à notre niveau, nous nous priverons de donner réponse à cette délicate interrogation; cependant, nous nous permettrons d'émettre nos réflexions néanmoins quant aux problèmes que posent aujourd'hui le partage des canaux de communications devant une demande sans cesse croissante et l'importance capitale que détient ce que les spécialistes nomment communément par la séparation aveugle des sources.

« Séparation aveugle des sources » est devenue et depuis relativement pas longtemps un terrain de prédilection des télécommunications et des disciplines faisant appel au traitement du signal en général. Cette expansion s'est accompagnée naturellement d'une exigence croissante dans les chaînes de traitements du signal, dans d'autres domaines tels que l'imagerie, le biomédical, l'acoustique, le contrôle non destructif, ... Cependant, force est de reconnaître qu'une des raisons fondamentales de son développement de nos jours est indubitablement l'essor des systèmes de communications radio-mobiles. Ce secteur a soulevé une zone non négligeable de nouvelles préoccupations inhérentes à son épanouissement. Exemple : Comment permettre à plusieurs mobiles d'émettre à la même fréquence et en même temps ? L'interrogation est des plus actuelles et peut être posée autrement : Comment et par quel mécanisme scientifique se fera la séparation du mélange convolutif des diverses émissions ?

Ces exemples sont parmi tant d'autres applications de la séparation aussi bien développées dans le secteur des télécommunications que dans le nucléaire, le médical, ou tant d'autres, en un mot dans le traitement d'antenne qui a pour objectif de retrouver des signaux sources ou de les localiser.

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction Générale

La connaissance scientifique s'est développée par un double effort : d'une part, la réflexion sur les mécanismes, c'est-à-dire sur la nature des interactions entre grandeurs physiques liées aux phénomènes; Cette réflexion se concrétise grâce à l'outil mathématique par les lois de la physique, relations abstraites entre grandeurs physiques; d'autre part, l'expérimentation qui repose sur la mesure des grandeurs physiques et qui, en leur associant une grandeur numérique, permet de définir quantitativement les propriétés des objets, de vérifier numériquement les lois physiques ou d'en établir empiriquement la forme.

Alors la science cherche à saisir puis à exprimer mathématiquement dans des théories cohérentes les lois régissant les rapports entre grandeurs physiques, la technique utilise ces lois et les propriétés de la matière pour créer de toute pièce des dispositifs ou des matériaux nouveaux qui permettent à l'homme d'accroître ses moyens d'actions afin de mieux assurer sa subsistance, de faciliter ses échanges et de réduire sa peine.

Le signal et l'information qu'il véhicule, son traitement, dans cette architecture en pleine transformation, occupent indubitablement une place privilégiée.

Si, dans un premier temps, la technique fût un recueil de procédés empiriques, fruits de l'observation, de tâtonnements aléatoires ou d'essais successifs, la connaissance des lois de la nature a permis à la technique de rationaliser sa démarche et de devenir une science de réalisation. La mesure y joue dès lors un rôle capital. La construction d'une machine ou la mise au point de matériaux nouveaux exigent de donner à leurs éléments constitutifs des caractéristiques que la mesure permet d'ajuster aux valeurs appropriées. Le fonctionnement d'une machine ou d'un appareillage doit être contrôlé afin que soient assurées la qualité des fabrications et la sécurité des hommes et des installations : or, contrôler c'est d'abord vérifier par la mesure qu'un certain nombre de grandeurs physiques ont les valeurs assignées.

Dans les laboratoires de recherche scientifique comme dans les installations industrielles, l'une des tâches principales du chercheur comme de l'ingénieur est donc d'effectuer des mesures des grandeurs physiques variées qui déterminent leurs expériences ou conditionnent le déroulement correct de leurs fabrications.

Des domaines aussi vastes, aussi inconnus, et le plus souvent aussi appréhendés s'en sont développés. « Le besoin crée la science » disait l'autre. Et s'inscrivant dans cette optique, est-on ramené aujourd'hui à parler de séparation de source. Et ceci ne relève en rien du hasard ou de l'extraordinaire si les plus grands travaux de recherche du moment s'y identifient.

En effet, la séparation aveugle de sources est un problème nouveau en traitement du signal. Elle doit son apparition à une approche de la neurobiologie des mouvements où la séparation des informations (étirement et vitesse d'étirement) d'un muscle se fait en aveugle du système nerveux central. Cette approche a donné la conviction qu'une solution à ce problème existait. Les progrès accomplis et les travaux en cours dans ce domaine ne cessent de démontrer son omniprésence actuelle dans, aussi bien, le champ théorique que pratique et de là même ouvrent consciemment ou inconsciemment d'autres portes à une infinité d'interrogations et surtout d'applications aussi potentielles que stratégiques.

L'importance de la séparation des sources nous paraît déterminante en conséquence puisque pouvant jouer un rôle fondamental dans la majorité des caractéristiques et élaboration d'une chaîne de traitement du signal (traitement de la parole, de l'image, du signal radar, ...). Le travail que l'on se propose d'accomplir au terme de ce mémoire cible effectivement ce que les spécialistes en signal appellent communément le visu d'un algorithme, autrement dit, sa matérialisation et avec cette nouvelle ère d'informatisation et de miniaturisation des systèmes, sa traduction physique en termes d'électronique moderne, numérique, non volumineuse et aussi flexible que possible : Sélectivement la technologie FPGA sera sollicitée en nous aidant d'outils pour le moins aussi novateurs que les langages de description hardware, spécifiquement avec le VHDL

Énoncée dans un autre langage, nous pouvons comprendre que la tâche qui nous est assignée est de « écrire » un algorithme de séparation aveugle de sources, en l'occurrence ici SOBI (Second Order Blind Identification), en VHDL.

En tant que nouveaux designers, nous allons tenter de satisfaire à cette exigence.

Aussi, l'architecture de ce mémoire a été bâtie selon deux grandes structures :

Une première qui élucidera la problématique de l'identification autodidacte de sources pour servir d'introduction à une étude exhaustive de l'algorithme SOBI.

Et une seconde qui se veut d'être l'acheminement prévu de la première en fournissant sa traduction hardware tout en assurant au préalable une dislocation des vocables VHDL, FPGA

Dans les prochaines lignes qui vont suivre, nous avons essayé d'apporter notre humble contribution, qui, espérons-le, sera serviable à toute la communauté scientifique.

Chapitre 1

Qu'entendre par séparation aveugle de sources?

I.1. Introduction	3
I.2. Historique.....	3
I.2.1. Les différentes approches.....	3
Les deux écoles	3
I.2.2. Les résolutions.....	4
I.2.2.a. Les algorithmes non-linéaires	4
I.2.2.b. Les algorithmes dans un environnement réel.....	5
I.2.3. Les travaux actuels.....	5
I.3. Position du problème	6
I.3.1 Les méthodes de résolution.....	6
I.3.2. Première définition	7
I.4. Résumé.....	9
I.5. Conclusion.....	9.

I.1. Introduction

Le premier pas que l'on empruntera sera de placer notre étude dans son contexte; Et seulement ensuite d'en dresser le fond de la problématique dans une seconde partie.

I.2. Historique

Le but d'il y a vingt ans est toujours resté le même avec quelques variantes mais l'objectif principal est de récupérer des sources indépendantes inconnues seulement par des observations qui sont issues de mélanges linéaires (ou non-linéaires) inconnus eux aussi des signaux sources.

I.2.1. Les différentes approches

D'emblée, force est de reconnaître que l'étude de la séparation autodidacte de sources à partir d'observations d'une rangée de capteurs a toujours été un problème classique et difficilement solvable dans le traitement des signaux.

Deux écoles de scientifiques ont eu à traiter de ce problème : ceux relevant du « domaine », c'est-à-dire les spécialistes en traitement des signaux, et d'un autre côté, ceux de la théorie de l'information.

Les deux écoles

Les premières approches furent corrélatives c'est-à-dire, grossièrement, cherchaient à traiter des fonctions du temps qui portaient une mesure possible du degré de ressemblance ou de dissemblance des divers signaux. Contrairement aux transformations basées sur les corrélations telle que la « Principal Component Analysis » (PCA), la séparation de source, non seulement décorrèle les signaux (statistiques du 2nd ordre), mais également réduit les dépendances statistiques d'ordres plus élevés, en essayant de rendre les signaux aussi indépendants que possible. L'objectif est d'exécuter une transformation linéaire qui rende les variables résultantes aussi statistiquement indépendantes les unes des autres que possible.

Deux communautés de recherche ont considéré cette analyse sous des approches différentes.

Les premiers travaux sur la séparation aveugle ont été menés par Héroult et Jutten (1986) qui ont introduit un algorithme adaptatif dans une architecture simple rétroactive qui pouvait séparer plusieurs sources indépendantes inconnues. Leur approche a été encore plusieurs fois améliorée : cinq ans plus tard en 1991 par eux-mêmes, puis par Karhunen et Joutsensalo (1994), Cichocki, Unbehauen et Rummert (1994). Comon (1994) introduisit le concept de « l'Analyse Composante Indépendante » et propose des fonctions liées à la minimisation approximative de l'information mutuelle entre les capteurs.

En parallèle, des règles basées sur la théorie de l'information ont été proposées par Linsker (1992). L'objectif -sans pour autant défier ou contrer celui de Comon, le rapprochait en fait- était de maximiser l'information réciproque entre les entrées et les sorties d'un réseau. Cette approche est liée au principe de la réduction de la redondance suggérée par Barlow (1961) comme stratégie de codage dans des neurones. Chaque neurone devrait coder les dispositifs

qui sont aussi statistiquement indépendants que possible des autres neurones sur un ensemble d'entrées; la décorrélation comme stratégie de traitement a été explorée par Atick (1992). Nadal et Parga (1994) ont prouvé que dans le cas de faibles perturbations, le maximum d'information réciproque entre l'entrée et la sortie d'un réseau neurologique impliquait que la distribution de la sortie était factorielle; c'est-à-dire, la fonction densité de probabilité peut être factorisée en un produit de densités marginales. Roth et Baram (1996) et Bell et de Sejnowski (1995) ont aboutit aux mêmes résultats pour cette maximisation. Leurs applications ont respectivement servi à l'analyse des séries chronologiques, et à la séparation aveugle des sources. Bell et Sejnowski (1995) ont mis le problème de la séparation aveugle de source dans un cadre purement relevant de la théorie de l'information et ont démontré avec cela la séparation et la déconvolution de sources mélangées. Leurs méthodes adaptatives appliquées sont plus plausibles pour une perspective de traitement que les fonctions basées sur les cumulants proposées par Comon (1994). Cardoso et Laheld (1996) ont proposé une méthode adaptative semblable pour la séparation de source.

I.2.2. Les solutions

D'autres algorithmes proposés pour la séparation et selon différents points de vue ont vu le jour. Des approches par la méthode de l'Estimation du Maximum de Vraisemblance (MLE) ont été proposées par Gaeta et Lacoume (1990) et élaborées pour la première fois par Pham (1992). Pearlmutter et Parra (1996), MacKay (1996) et Cardoso (1997) ont prouvé que l'approche d'Infomax de Bell et de Sejnowski (1995) et l'approche du l'Estimation du Maximum de Vraisemblance sont équivalentes. Girolami et Fyfe (1997), motivé par des index théorétiques, pour le compte d' Exploratory Projection Pursuit (EPP) utilisèrent l'entropie relative comme index de projection et prouvèrent que la poursuite de la recherche de la projection de Kurtosis extrairait une des sources fondamentales à partir d'un mélange linéaire. Un réseau multiple du modèle EPP a été développé pour permettre la pleine séparation de toutes les sources inconnues (Girolami et Fyfe, 1997).

I.2.2.a. Les algorithmes non-linéaires

D'autres algorithmes qualifiés de non-linéaires développés par Karhunen et Joutsensalo (1994), Xu (1993) et Oja (1997) peuvent également être tirés du principe d'Infomax puisqu'ils réduisent au minimum approximativement la somme des carrés des cumulants marginaux de quatrième ordre (Comon, 1994) et donc réduisent au minimum l'information mutuelle des observations (Girolami et Fyfe, 1997).

Bell et Sejnowski (1995) ont précisé une similarité entre leur algorithme d'infomax et celui de Bussgang dans le traitement des signaux; et c'est Lambert (1996) qui élucida le lien entre trois différentes fonctions de contraste de Bussgang. Lee et autres, plutard(1998) montreront comment la propriété de Bussgang se relie au principe d'infomax et comment toutes ces approches apparemment différentes peuvent être mises dans un cadre d'unification pour le problème de séparation de source basé sur une approche théorétique de l'information.

La méthode originelle d'Infomax pour la séparation aveugle de Bell et de Sejnowski (1995) convenait beaucoup plus aux sources super-gaussiennes. Aussi, Girolami et Fyfe (1997), déduisent, par le choix de l'entropie relative comme index de poursuite de projection, une règle d'étude qui peut séparer aveuglément des distributions mélangées de sources sous et super-gaussiennes. Lee, Girolami et Sejnowski (1997) prouvent que leur méthode est une prolongation du principe d'infomax satisfaisant un critère général de stabilité et préservan:

l'architecture simple de Bell et de Sejnowski (1995). Les simulations et les résultats sur des données physiologiques réelles montrent la puissance des méthodes proposées (Lee, Girolami et Sejnowski, 1997).

Des simulations plus étendues ont été effectuées pour démontrer la puissance de l'algorithme d'étude.

Cependant, concernant *le mélange instantané*, le défi se déclare en faisant face à de vraies données réelles. Jung et autres (1997) prouvent que l'algorithme prolongé d'infomax peut décomposer linéairement des objets façonnés d'EEG tels que le bruit de ligne, l'œil clignotant, et bruit cardiaque avec des distributions sous et super-gaussiennes. McKeown et Al (1997) ont employé l'algorithme amélioré pour étudier l'activité du cerveau humain avec des données fMRI. En déterminant les régions du cerveau qui contiennent des quantités significatives composants indépendants temporellement, ils pouvaient indiquer la distribution spatiale des activités du cerveau en fonction des tâches. D'autres applications potentielles peuvent résulter en explorant les indépendances dans des images. Bell et Sejnowski (1997) proposent que les composants indépendants des scènes normales soient des filtres à faible bande. Ces filtres sont localisés et la plupart du temps orientés et semblables à ceux de Gabor. Les sorties des filtres sont peu abondamment distribuées. Bartlett et Sejnowski (1997) et Gray, Movellan et Sejnowski (1997) démontrent l'utilisation réussie des filtres de séparation comme dispositifs dans des tâches d'identification du visage et des tâches de lipreading .

I.2.2.b. Les algorithmes dans un environnement réel

Pour les applications précédentes, le modèle de mélange instantané peut être approprié parce que les retards de propagation sont négligeables. Cependant, dans des environnements réels où des retards assez conséquents peuvent se produire, une architecture et un algorithme deviennent alors nécessaires pour expliquer le mélange des sources retardées et des sources convoluées.

Ce problème de la séparation de source trouvera une réponse avec Yellin et Weinstein (1994) et avec Ngyuen et Jutten (1995) ainsi que d'autres basée sur des critères de cumulants d'ordre quatre.

Cependant, un solutionnement du problème en ne faisant appel qu'aux statistiques du deuxième ordre est possible et a été introduit par Bélouchrani et Al (1995). A la clé, un blanchiment des données et une diagonalisation conjointe des matrices d'intercovariances des observations.

Une extension du point de vue d'infomax employant une architecture de rétroaction a été développée par Torkkola (1996). Lee, Bell et Lambert (1997) ont prolongé le problème de la séparation de source en un système de feedback et un système de feedforward. L'architecture feedforward permet l'inversion des systèmes qui ne sont pas à phase minimale. En outre, les règles sont prolongées en utilisant des filtres avec l'algèbre matriciel dans le domaine fréquentiel (Lambert, 1996). La méthode proposée peut avec succès séparer des voix et la musique enregistrées dans un environnement réel. Lee, Bell et Orglmeister (1997) prouvent également que le taux d'identification d'un système automatique de reconnaissance de la parole est augmenté après séparation des sons articulés.

I.2.3. Les travaux actuels

Bien qu'efficace, cette randonnée à la recherche de l'indépendance des parties composantes du mélange linéaire est, avec toute objectivité, assez restreint et plusieurs chercheurs ont commencé à avouer quelques limitations.

Une prolongation évidente mais non triviale est le modèle de mélange non-linéaire. Dans (Hermann et Yang, 1996; Lin et Cowan, 1997; Pajunen, 1997) , les sources composantes non-linéaires sont extraites en utilisant les self-organizing-feature-maps (SOFM). D'autres chercheurs (Burel, 1992; Lee, Koehler et Orglmeister, 1997; Taleb et Jutten, 1997; Yang, Amari et Cichocki, 1997) ont employé une amélioration plus directe aux modèles précédemment présentés. Ils incluent certaines non-linéarités flexibles dans le modèle de mélange et le but est d'inverser la matrice de mélange linéaire aussi bien que le mélange non-linéaire. Plus récemment, Hochreiter et Schmidhuber (1998) ont proposé un codage et un décodage pour des mélanges non-linéaires.

D'autres limitations telles que le problème de l'identifiabilité dans la séparation, c'est-à-dire par exemple.. en ayant moins de capteurs que des sources et des modèles de bruit dans la formulation sont sujettes à des efforts courants de recherches.

La séparation aveugle de source est une méthode assez nouvelle et généralement s'adressant à plusieurs défis dans le traitement des signaux. Elle pointe une diversité des questions théoriques et ouvre une variété d'applications potentielles. Les récents résultats réussis dans des systèmes d'EEG, de fMRI, de reconnaissance de la parole et de restitution d'image indiquent la puissance et l'espoir optimiste dans ce nouveau paradigme.

I.3. Position du problème

Sommairement, par la séparation autodidacte de sources, il a été convenu d'entendre la restitution de ces signaux en absence à priori de toute information sur la façon dont ceux-ci ont été mélangés. La théorie veut qu'à partir d'une antenne de m capteurs mélangeant linéairement et instantanément n signaux sources obéissant à la loi de l'interindépendance statistique, on recueille et on retrouve ces signaux initiaux à un facteur près. Et tout le travail se fera exclusivement en ne faisant appel qu'aux statistiques des signaux sources; Suivant l'exploitation de ces données, différentes approches de résolution du problème de la séparation aveugle ont été développées.

I.3.1 Les méthodes de résolution

Si la cohérence temporelle est observée entre les signaux sources,

$$\text{C'est-à-dire } \begin{cases} \Gamma_{s_i} = E[s_i(t)s_i^*(t-\tau)] \neq 0 & \text{lorsque } \tau \neq 0 \\ \Gamma_{s_i}(0) \neq 0 \end{cases} \quad \forall i = 1, 2, 3, \dots, n.$$

✓ On exploite les statistiques d'ordre deux des signaux.

* Si les signaux sources sont temporellement blancs ou

* Si les signaux sources présentent des densités spectrales normalisées identiques

Et en absence d'informations à priori sur la distribution des sources,

- ✓ L'appel aux statistiques d'ordre supérieur à deux des signaux reçus est incontournable .
- Et si la distribution des signaux est connue,
- ✓ La possibilité de résolution avec la méthode du Maximum de Vraisemblance nous est offerte .
- ✓ Si les signaux sources sont temporellement blancs et gaussiens,
- Alors la séparation aveugle devient insolvable;

Le travail qui nous a été demandé traite exclusivement de la séparation au second ordre .
 La séparation des sources aux vues des résultats fournis de nos jours dans le traitement d'antenne surtout en bande étroite est des plus satisfaisants. Le filtrage d'antenne fait partie des méthodes de traitement des signaux *multidimensionnels* où, à la différence des méthodes monodimensionnelles, plusieurs capteurs, géographiquement, ou encore spatialement différents collectent les données à traiter. Limité au contexte des communications, où il s'agit de recevoir, dans les meilleures conditions, un signal utile (portant une information sensible), le filtrage d'antenne consiste à déterminer un traitement, linéaire la plupart du temps, associant au signal vectoriel un scalaire, qui maximise, en sortie de traitement, le rapport signal sur bruit. La localisation de sources, quant à elle fait aussi partie intégrante, au sens large, du filtrage d'antenne, on la rencontre néanmoins presque exclusivement sous le vocable localisation, dans la mesure où elle se « nourrit » des mêmes types de signaux, mais est destinée, en général, à déterminer la position angulaire, l'éloignement et la puissance des sources rayonnantes, sans connaissance particulière concernant la forme des signaux qu'elles émettent .

1.2.2. Première définition

On serait tenté d'émettre certaines conclusions hâtives quant à la séparation autodidacte de sources si l'on se contente d'interpréter les définitions, données par le dictionnaire, des mots

☞ séparation (action ,d'isoler, de séparer: faire un tri parmi des objets présentant un caractère différent): qui traduit comme préalablement souligné la restitution de signaux au niveau d'un mélange particulier.

☞ Autodidacte (qui s'instruit lui-même) :le travail se fera avec comme moyen :les statistiques de ces mêmes signaux à séparer .

et de

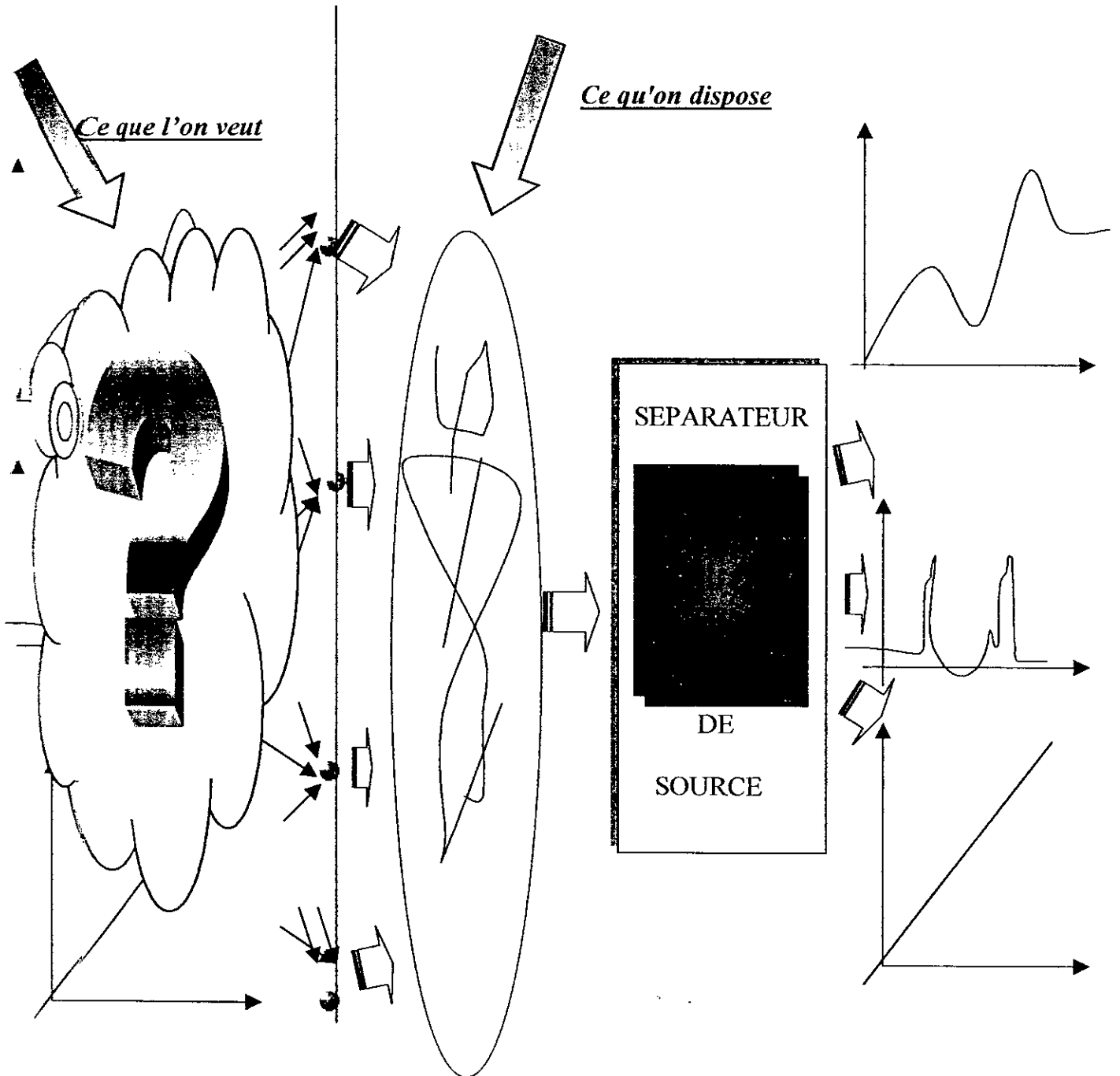
☞ sources (systèmes qui sont à l'origine de quelque chose) :qui peuvent être un signal ou une image .

On obtient déjà une idée précise de ce que les spécialistes entendent aujourd'hui par la juxtaposition des trois vocables; mais en plus de cette opération de restitution, la séparation aveugle participe également et efficacement dans la calibration d'antenne par une estimation des vecteurs directionnels des sources.

Cependant, nous ne pouvons nous satisfaire de ces approches assez peu littéraires bien que l'idée de base soit ainsi jetée; Nous verrons plus loin que les concepts d'identifiabilité et d'identification perturberont sensiblement ces définitions et précisement certaines spécificités liées à la fois à la nature et à la disponibilité de la banque d'informations à exploiter, alliées aux exigences de traitement, y changeront quelque chose. En effet, il est clair qu'une restitution ou une séparation complète défiera toujours l'algorithme à utiliser ou son implantation et ce dans n'importe quel environnement.

Pour synthétiser tout ce qui vient d'être dit, nous présentons ce petit schéma explicatif .

Fig 1-1 : Schéma simplifié de la séparation de sources



I.3. Résumé

Même si le premier algorithme a été mis au point il y a pas moins de vingt ans, il n'en demeure pas moins que la séparation aveugle ait toujours été un problème depuis l'aube de l'humanité; Ces concepts et techniques présentés ci-dessus sont les plus récemment développés dans ce domaine. Cependant, toutes ces techniques peuvent être regroupés dans deux catégories : Les HOS et Les SOS.

*** Les méthodes dérivées de HOS (Statistique d'ordre supérieur)**

Il y a trois sous-groupes dans cette catégorie :

Les méthodes dérivées du modèle caché de Markov(HMM), les méthodes polyspectrales et les méthodes de Bussbang.

1. Les méthodes dérivées de HMM

Utilisées lorsque tous les échantillons du signal donnent un alphabet fini et avec la supposition que la fonction de transfert du système soit causal pour une durée finie. Les propriétés markoviennes interviennent dans la maximisation de l'espérance (EM).

Inconvénient : nécessité d'un grand espace mémoire et des calculs complexes pour le processus markovien.

2. Les méthodes polyspectrales

C'est l'exploitation des spectres d'ordres élevés pour l'estimation première du canal et ensuite approximer les sources.

3. Les méthodes de Bussbang

Ces méthodes procèdent par l'égalisation du canal en minimisant une fonction coût qui contient des informations des moments d'ordre supérieur du signal de sortie.

L'inconvénient majeur des méthodes HOS réside dans la lenteur de la convergence, dans leur nécessité d'une taille d'échantillons plus grande pour avoir les approximations moyennes de HOS avec exactitude; par conséquent, ces méthodes ne peuvent être appliquées dans des applications où les variations du canal sont très rapides.

*** Les méthodes SOS**

Méthodes de la matrice paire, de cross corrélations, de LP(Linear Prediction), ML (Maximum likelihood), du domaine fréquentiel

I.5. Conclusion

Dans les prochaines lignes qui vont suivre, nous allons essayer autant que possible d'homogénéiser toutes ces approches en introduisant les différentes notions de mélanges instantanés, d'identification, de séparation ainsi qu'en y tout l'arsenal mathématique et probabiliste indispensable. En une question : Comment notre algorithme de base, en l'occurrence SOBI a opéré pour résoudre ce problème ?

Mais avant cela, nous nous permettrons de glisser quelques rappels essentiels.

Chapitre 2

Rappels

II.1. Introduction.....	10
II.2. Les capteurs	10
II.2.1 Définition	10
II.2.2 Les principaux types de capteurs.....	10
II.2.3 Grandeurs d'influence.....	11
II.3 Les sources	12
II.3.1. Définition.....	12
II.3.2. Les types de sources.....	12
II.3.2.a. Source analogique	12
II.3.2.a. Source discrète	12
II.3.3. Exemples.....	12
II.4. Conclusion.....	13

II.1 Introduction

Détecter un signal utile ou une information consiste à prendre une décision fondée sur un faisceau de présomptions déduites de l'analyse des propriétés d'un vecteur d'observations, issu d'un système d'un ou d'un système de capteurs tel dans la séparation de source. Et précisément dans le cas de la séparation des sources, c'est là où ce premier maillon de la chaîne de traitement constitué par ces capteurs, joue un rôle assez fondamental.

II.2. Les Capteurs

II.2.1 Définition

Afin d'être menée à bien, l'opération de mesure nécessite généralement que l'information qu'elle délivre soit transmise à distance du point où elle est saisie, protégée contre l'altération par des phénomènes parasites, amplifiée, avant d'être exploitée de diverses manières : affichée, enregistrée ou traitée par ordinateur. L'Electronique offre à cet égard des moyens divers et puissants : pour en tirer le meilleur parti et qu'en bénéficient les mesures de tous types de grandeurs physiques, comme leur traitement et leur exploitation, il est très souhaitable de transformer immédiatement sous la forme d'un signal électrique chacune des grandeurs physiques intéressantes. C'est le rôle du capteur que d'assurer cette duplication de l'information en la transférant, au point même où se fait la mesure, de la grandeur physique (non électrique) qui lui est propre, sur une grandeur électrique : courant, tension, charge ou impédance.

Un capteur est d'abord le résultat de l'exploitation ingénieuse d'une physique.

Un des problèmes importants dans la conception et l'utilisation d'un capteur est la constance de sa sensibilité qui doit dépendre aussi peu que possible :

- de la valeur du mesurande (source) et de sa fréquence de variation (bande passante)
- du temps (vieillessement)
- de l'action d'autres grandeurs physiques de son environnement qui ne sont pas l'objet de la mesure et que l'on désigne comme grandeur d'influence.

II.2.2 Les principaux types de capteurs

La diversité des capteurs est due à la combinaison de la multitude des grandeurs à mesurer de la multitude des principes de mesure utilisés et des nombreuses conditions d'environnement et d'implantation, suivant les applications. On peut donner à titre indicatif l'existence des:

- ☞ Capteurs optiques
- ☞ Capteurs de température
- ☞ Capteurs de position et déplacement
- ☞ Capteurs de déformation
- ☞ Capteurs tachymétriques
- ☞ Capteurs de force
- ☞ Capteurs d'accélération.
- ☞ Capteurs de vitesse

- ☄ Capteurs de pression de fluide.
- ☄ Capteurs de mesure de vide.
- ☄ Capteurs acoustiques
- ☄ Capteurs de rayonnements nucléaires
- ☄ Capteurs électrochimiques
- ☄ Capteurs de composition gazeuse

II.2.3 Grandeurs d'influences

Un capteur, de par ses conditions d'emploi, peut se trouver soumis non seulement au mesurande mais à d'autres grandeurs physiques dont les variations sont susceptibles d'entraîner un changement de la grandeur électrique de sortie qu'il n'est pas possible de distinguer de l'action du mesurande. Ces grandeurs physiques «parasites» auxquelles la réponse du capteur peut être sensible sont les grandeurs d'influence.

Ainsi, par exemple: la température est grandeur d'influence pour un capteur optique comme la résistance photoconductrice; il en est de même pour le champ magnétique vis-à-vis d'un capteur thermométrique comme la résistance de germanium.

Les principales grandeurs d'influence sont:

- La température
- La pression, l'accélération et les vibrations
- L'humidité
- Les champs magnétiques variables ou statiques
- La tension d'alimentation

Afin de pouvoir déduire de la mesure de x , les observations les valeurs de s représentant les sources, il est donc nécessaire de:

- ✓ Soit de réduire l'importance des grandeurs d'influence au niveau du capteur en le protégeant par un isolement adéquat : supports antivibratoires, blindages magnétiques.
- ✓ Soit de stabiliser les grandeurs d'influence à des valeurs parfaitement connues et d'étalonner le capteur dans ces conditions de fonctionnement : enceinte thermostatée ou à hygroscopie contrôlée, sources d'alimentation régulées ;
- ✓ Soit enfin d'utiliser des montages qui permettent de compenser l'influence des grandeurs parasites : pont de Wheatstone avec un capteur identique placé dans une branche adjacente au capteur de mesure.

Le capteur, premier élément de la chaîne de mesure est la source déterminante du signal électrique que le reste de la chaîne doit traiter et exploiter. C'est donc de la qualité que dépendent en premier lieu d'une part, la plus ou moins concordance entre valeur mesurée et valeur vraie du mesurande et d'autre part, les limites de l'incertitude sur la valeur mesurée.

II.3 Les sources

II.3.1 Définition

La source est un mécanisme servant à choisir, parmi l'ensemble de messages possibles et d'une manière imprévisible pour l'observateur, un certain message particulier destiné à être transmis à un correspondant. Les réalisations particulières de la source s'appellent *symboles*

II.3.2 Les types de sources :

II.3.2.a. Source discrète

Une source est dite discrète si elle présente un alphabet fini.

Une source est dite discrète si elle produit de l'information à partir d'un nombre fini n de caractères (lettres, chiffres, signes, etc) qui, par leurs diverses combinaisons, constituent des messages. L'information ainsi générée est de nature numérique (digitale).

L'ensemble des n caractères disponibles constitue, par analogie avec la langue écrite qui est une source typiquement discrète, un alphabet.

II.3.2.b. Source analogique

Une source est dite continue si elle présente un alphabet infini.

Si la source est continue en ce sens qu'elle exprime des nuances d'une infinie subtilité, l'information produite est analogique.

II.3.3 Exemples

II.3.3.a. Source analogique : la musique

La musique est une source d'information continue, elle se distingue par :

- un spectre moyen à long terme plus étendu vers les hautes fréquences
- des basses fréquences dès 15HZ
- une distribution statistique moins structurée dans le temps
- des pauses moins fréquentes.

II.3.3.b. Source discrète

Un texte est un document constitué de caractères (lettres, chiffres, signes, etc.) issus d'un alphabet fini il contient donc une information numérique, produite par une source discrète, selon les règles de la sémantique (signification des mots) et de la syntaxe (grammaire) propre à une langue.

L'alphabet latin minimum comprend :

26 lettres

10 chiffres arabes

l'espace entre deux mots

quelques signes de ponctuation

C'est l'alphabet utilisé par le télégraphe et le télex.

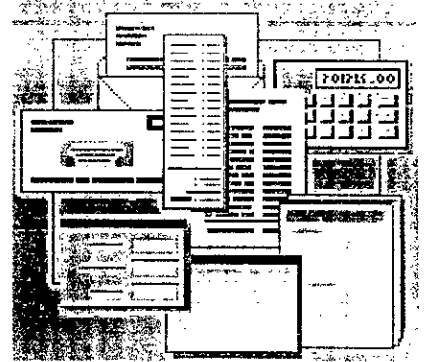
II.4 Conclusion

Dans cette étude, il s'agira surtout de question de traitement des signaux à la sortie de capteurs.

La séparation aveugle et principalement *SOBI*, *n'exige aucune différenciation au niveau du capteur, ni de sa nature, et de son adéquation*, cependant il est à noter le rôle central que peut jouer cet élément dans la chaîne de traitement et des conséquences sur nos résultats pour un quelconque dysfonctionnement à son niveau.

Même remarque pour nos sources sinon qu'elles doivent vérifier certaines conditions statistiques.

Chapitre 3



Méthodologie de traitement

III.1. Introduction	14
III.2. Les Mélanges instantanés	14
III.3. Notions d'identifiabilité et d'identification	16...
III.3.1. L'identifiabilité	16
III.3.2. . L'identification	17
III.3.2. . La séparation	18
III.4. Conclusion	18

III.1. Introduction

Les idées fondamentales étant jetées, l'analyse proposée ne se veut, d'aucune façon, exhaustive. Son découpage, reprend, lorsque cela est possible, les subdivisions apparentes de la première référence citée [Bélou 1]. En effet, cette dernière restera logée à l'intérieur des rubriques qui auront à suivre tout au long de ce travail, dans la mesure où leur vocation principale rejoignent, dans l'intention, nos préoccupations premières, qui ne sont autres et principalement, que la traduction de SOBI en VHDL. Introduisons d'abord ce qui a été nommé par :

III.2. Les Mélanges instantanés

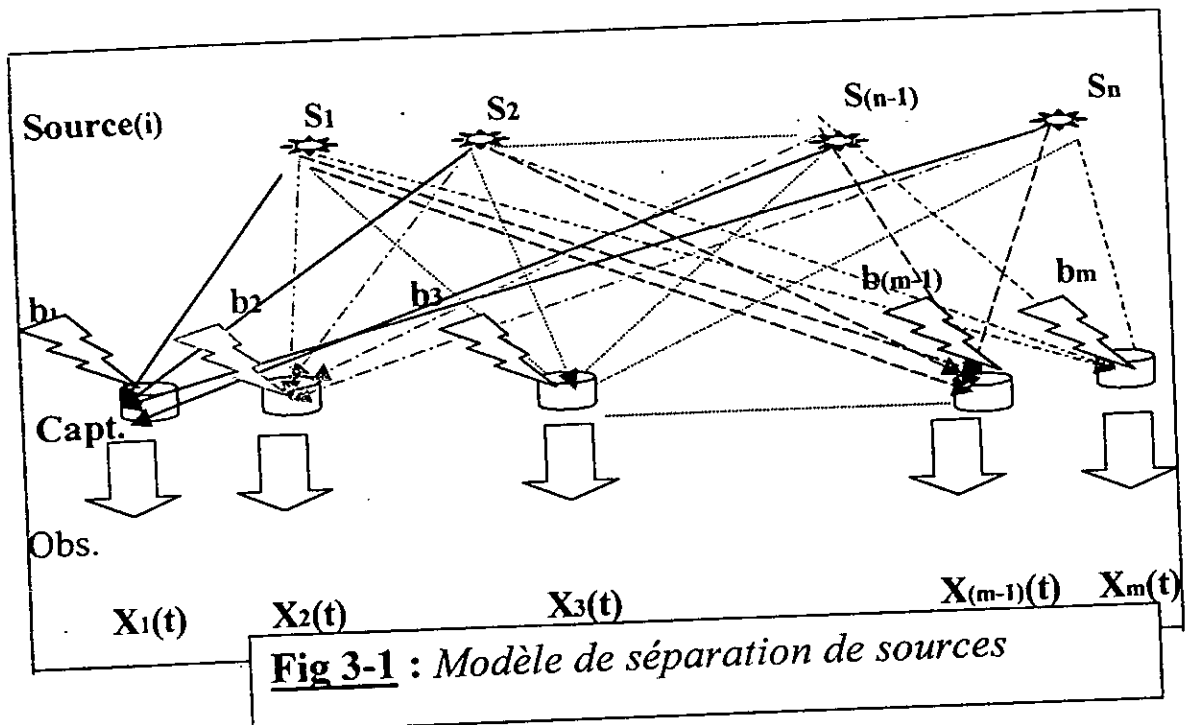
Eu égard au contenu de l'introduction, la première tâche qui incombe au concepteur consiste à élaborer un modèle suffisamment exploitable pour couvrir une grande variété de signaux et de mise au point, malgré tout, aisée, en vue d'une réalisation de complexité réduite. La modélisation proposée dans ce travail, est celle d'un canal où on observe un mélange instantané et linéaire des signaux sources.

Schéma de la séparation

La nature physique des phénomènes dont les signaux sont les prolongements influe par voie de conséquence sur le choix des méthodes destinées à les déterminer. Si l'on étudie un signal pendant une durée inférieure à ce que l'on appelle son intervalle de stationnarité (laps de temps sur lequel le signal exhibe une certaine stationnarité (à titre illustratif, cette durée est de 10ms pour un signal de la parole), l'analyse spectrale est la méthode correspondante. Donc nous aurons affaire avec des signaux stationnaires d'abord. Le modèle adopté pour le canal de transmission est un modèle simplifié en ce sens que les phénomènes de propagation autre que l'affaiblissement en espace libre, qui influe sur le rapport signal sur bruit, ne sont pas pris en compte. Le canal est supposé stationnaire, c'est-à-dire que ses caractéristiques ne varient pas dans le temps ; le problème de l'auto-adaptativité du canal n'est pas donc abordé.

En second lieu, nous traiterons de signaux à valeurs complexes. En effet, la complexification d'un signal (passage d'un original à valeurs réelles à valeurs complexes s'avère indispensable dans les applications de nature très différentes, allant de la suppression des interférences entre les composants de fréquences négatives et positives dans certaines représentations aux télécommunications bande étroite (ce qui pourrait nous concerner le plus). En tant qu'opération de codage, elle requiert un minimum d'arbitraire et un maximum de rendement. Elle vise, somme toute, à généraliser à des signaux quelconques, l'opération très classiques qui consiste à associer une exponentielle à un cosinus. Si la complexification est une opération de compression de l'information, elle vise à supprimer une forme de redondance. Donc en résumé, en séparation des sources, on s'intéresse plutôt à des signaux aléatoires stationnaires et complexes. Par ailleurs, l'hypothèse fondamentale sur le bruit est son caractère gaussien et additif.

Le modèle de séparation suppose qu'à l'instant t , nous avons le schéma de la figure ci-après :



Avec un nombre de capteurs m supérieur ou au moins égal à celui des sources n (nous verrons ultérieurement pourquoi), nous observons des mélanges linéaires instantanés au niveau de chaque capteur; On peut alors écrire, à la sortie :

$$(1) \begin{cases} X_1(t) = a_{11} s_1(t) + a_{12} s_2(t) + \dots + a_{1n} s_n(t) + b_1(t) \\ X_2(t) = a_{21} s_1(t) + a_{22} s_2(t) + \dots + a_{2n} s_n(t) + b_2(t) \\ \vdots \\ X_m(t) = a_{m1} s_1(t) + a_{m2} s_2(t) + \dots + a_{mn} s_n(t) + b_m(t) \end{cases}$$

Notre équation de base se transforme aussi :

$$(1) \iff X_i(t) = \sum_{j=1}^n \sum_{k=1}^m a_{ij} s_j(t) + b_k(t) \quad (2)$$

Première indétermination fondamentale mais qui a son avantage également puisqu'il permettra de normaliser le problème; la dynamique de nos signaux peut être sujette à la supposition sans perte de généralité cependant. *La puissance de chaque signal sera supposée égale à 1*, autrement dit, les signaux sources seront de variance unitaire. Conséquence immédiate: leur autocovariance sera la matrice identité I .

Deuxième indétermination: l'indice i est arbitrairement attribué à une source, soit donc les sources seront déterminées à une permutation près.

En définitive, séparation autodidacte de sources signifie obtention de signaux proportionnels aux signaux sources recherchées et non-restitution directe et fidèle comme précédemment émis; En outre le mélange ne sera connu qu'à une phase et permutation près.

Une dernière remarque à noter également explique le choix du mélange linéaire instantané, qui sans pour autant constituer une condition sine qua non d'identification, n'en est pas moins intéressante; En effet, c'est grâce à ce choix que les processus de blanchiment et d'unitarisation des matrices de covariance seront indépendantes du retard. Nous verrons tout ceci plus en détails dans les prochains chapitres.

Ailleurs, on ne pourrait clore cette partie sans répéter encore cette autre condition de l'identification qui repose sur la disposition spatiale et le nombre de capteurs. En effet ce dernier doit obligatoirement vérifier une relation de supériorité ou au moins d'égalité par rapport au nombre de sources à identifier. Le cas contraire causerait une incompatibilité de dimension lors du traitement spécialement dans des multiplications matricielles : *m doit être supérieur ou égal à n .*

III.3.1. L'Identification

L'identification, c'est l'opération, approche expérimentale ou non, de détermination des caractéristiques dynamiques d'un procédé (système) dont la connaissance est nécessaire pour la conception et la mise en œuvre d'un système performant. Elle comporte quatre étapes :

- ① Acquisition des entrées/sorties sous un protocole d'expérimentation
- ② Choix de la structure du « modèle » (complexité)
- ③ Estimation des paramètres du modèle
- ④ Validation du modèle identifié

L'identification et la modélisation sont deux concepts différents et se distinguent par une autre façon de poser un problème voisin. Cette manière spécifique émane du fait qu'en Identification, on dispose, en général, d'une connaissance supplémentaire d'un mécanisme physique à l'origine de la génération des données disponibles (les signaux sources dans le cas de la séparation et les informations aprioriques) qui révèle, assez précisément, une relation mathématique de cause à effet entre ces données et une autre grandeur, qualifiées parfois d'entrée, dont on cherche à se rapprocher le plus possible. Plus généralement, l'Identification s'impose comme un problème inverse qui, lorsque la relation de cause à effet, s'apparente d'une façon ou d'une autre (soit de façon exacte, soit moyennant un développement limité) à un filtrage linéaire invariant, repose une méthode privilégiée appelée déconvolution. Lorsque la matrice qui caractérise la relation linéaire discrétisée est connue, on parle de déconvolution standard. Si celle-ci n'est pas accessible, le problème reste quand même soluble sous réserve que le processus aléatoire observé est *non gaussien*. On a recours, à ce moment là, aux méthodes dites de séparation des sources et de déconvolution aveugle ou myope. Conformément à notre cas de figure, l'identification est dès lors applicable. Aussi, comme principaux outils, les spécialistes ont fait appel au filtrage spatial d'abord pour accomplir la

tâche d'estimation d'un signal source après connaissance des signaux capteurs et du mélange. En résumé, il s'agit détecter des faits expérimentaux inédits en se fondant uniquement sur les observations, soit l'estimation des quantités inhérentes à la modification des propriétés des signaux observés.

III.3.2. La séparation

Le mélange identifié, la phase de séparation peut être donc lancée ; Ici, également, deux cas de figures peuvent occurer :

- ① Selon qu'on veuille optimiser le rapport signal/bruit
- ② Selon qu'on veuille optimiser le rapport signal/interférences

Dans tous les deux cas, ce sera un filtrage spatial qui résoudra le problème de la séparation.

L'annuleur d'interférences répond aux spécifications de ② en fournissant une estimée de chaque signal source (un seul signal bruité) à condition que les colonnes de la matrice du mélange présentent une indépendance linéaire :

$$\hat{s}_i(t) = a_i^* R_y^\# x(t)$$

R_y = autocorrélation de la partie du mélange $y(t)$

Le filtre adapté spatial donne un signal source avec cependant une contribution des autres sources et du bruit selon le schéma :

$$\hat{s}_i(t) = a_i^* R_y^{-1} x(t)$$

III.4. Conclusion

Une fois que le système à traiter obéit à certaines règles minimales et que suppositions inévitables aient été faites, la technique de la séparation pouvait s'appliquer.

Aussi, il convient de noter également l'utilisation de la séparation adaptative où c'est un algorithme adaptatif non linéaire qui régule le tout et de la séparation par blocs à une ou à deux étapes.



Applications de la séparation aveugle de sources

IV.1. Introduction.....	19
IV.2. Les applications.....	19
IV.2.1. Les télécommunications.....	19
IV.2.2. La séparation d'images.....	19
IV.2.3. La reconnaissance de la parole.....	20
IV.2.4. Le traitement du signal sismique.....	20
IV.3. SOBI	20
Exemple d'une application de SOBI	21
IV.3. Conclusion.....	22

IV.1. Introduction

La puissance d'un algorithme se justifie en grande partie par ses applications pratiques et les ouvertures induites, c'est-à-dire son utilisation comme maillon dans d'autres chaînes de traitement ou bien même son intégration dans d'autres algorithmes.

IV.2. Les applications

En traitement du signal, les signaux observés à la sortie d'un capteur constituent un mélange de signaux élémentaires auxquels ce capteur est sensible, par exemple des signaux sonores à la sortie d'un microphone qui sont dus à plusieurs sources inconnues. Dans le cas de la séparation dite aveugle, les informations sur les signaux sources ne sont pas disponibles, ce qui rend l'opération plus difficile.

Ceci implique l'utilisation de l'identification aveugle dans une diversité d'applications et dans des domaines variés de la science.

Nous avons choisi de vous en exposer brièvement quatre: les télécommunications, le traitement d'images, le traitement de la parole, et le traitement du signal sismique.

IV.2.1. Les télécommunications

Dans ce champ, il est admis et démontré depuis longtemps que la réponse fréquentielle d'un canal inconnu de largeur de bande finie cause ce que l'on nomme par interférences intersymbôles (ISI). Donc, le travail qui se définissait alors serait d'éliminer ces interférences sans pour autant avoir une quelconque information sur les signaux sources !

La stratégie est qu'on passera à une identification du canal, mais « aveuglement ».

Cette technique est particulièrement présente en communication numérique et l'essor des systèmes radio-mobiles en a rendu son terrain de prédilection de nos jours. Ici, la séparation autodidacte s'affaira tout d'abord à l'identification du canal avec ses différentes caractéristiques et le tout en temps réel pour maintenir correctement le flux d'informations qui le traverse. Ce travail est accompli en faisant appel à une grande fraction de la capacité du canal à identifier pour un traitement en apprentissage; Dans les GSM, bien que les techniques d'accès multiples tels que le FDMA (Frequency Division Multiple Access) / TDMA (Time Division Multiple Access) / CDMA (Code Division Multiple Access) tendent à augmenter la capacité du canal de transmission, il reste toujours encore à minimiser cette fraction puisée pour la séquence d'apprentissage qui demeure considérable. Et c'est à ce niveau que l'exploitation de la séparation aveugle devient intéressante de la simple innovation que le récepteur identifiera et égalisera le canal seulement à partir des signaux reçus. On a plus besoin de séquence d'apprentissage dès lors.

En plus, à l'orée de ce nouveau millénaire de nouvelles techniques portant le nom de SDMA (Space Division Multiple Access) s'annoncent et la séparation aveugle y apportera sans nulle doute sa solution d'autant plus que l'on permettra à plusieurs mobiles d'émettre en même temps et à la même fréquence. La séparation pour des mélanges convolutifs peuvent alors prévaloir.

IV.2.2. La séparation d'images

Une image floue à la télé due au mouvement de la caméra, ou à un mauvais réglage (position du focus, lentilles légèrement divergentes ou convergentes, ..) est un exemple assez fréquent de la dégradation d'une image. D'autres, que l'on qualifierait de beaucoup plus « sérieux » relèveraient de l'astronomie par exemple, ou encore de la télédétection ou plus encore de l'imagerie médicale.

Si l'on considère que l'image originale constitue notre signal d'entrée et sa sortie la version floue ou mélangée, une approche de séparation d'identification pourrait être envisagée pour la restauration. En plus lorsque la fonction de transformation du système est inconnue (ce qui arrive le souvent dans la pratique exple : la matrice de calibration de la caméra), le traitement autodidacte s'impose.

Il est à noter, vu que nous sommes en train de parler de séparation au second ordre et plus précisément de SOBI, qu'un travail supplémentaire de passage d'un espace à deux dimensions vers un autre à une dimension est alors nécessaire pour pouvoir appliquer SOBI. (En Matlab, la fonction `reshape` sera alors utilisée).

IV.2.3. La reconnaissance de la parole

On a souvent besoin d'identifier tel ou tel son spécifique surtout en reconnaissance de la parole où commande et pilotage de machines en dépendent.

Aussi, des perturbations tel l'environnement d'enregistrement sont les plus fréquemment observables, et conduisent à des techniques aveugles.

IV.2.4. Le traitement du signal sismique

La séparation aveugle intervient également au niveau de la sismologie lorsqu'en exploration, des charges sont dynamitées et qu'un géophone est utilisé pour recevoir le signal réfléchi et/ou diffracté; L'utilité de ce signal réside dans l'estimation des coefficients de réflexion qui sont associés aux réponses impulsionnelles des différents rayons de la terre et l'éclaircissement de ses caractéristiques physiques. Ici, le signal reçu est représenté par tous les échos produits des différents rayons de la terre en réponse à l'excitation, qui, ordinairement, est sous la forme d'une pulsation de courte durée.

Et encore une fois, c'est un problème d'identification dû au fait que la forme d'onde exacte de l'excitation responsable de la génération du signal reçu est très souvent inconnue.

SOBI, à l'instar des algorithmes de séparation de source a subi différentes implémentations et conduit à diverses applications.

IV.2. SOBI

Sa première implémentation à titre d'expérimentation fût en MATLAB. Le programme est sauvegardé dans un fichier SOBI.m, code de paramètres :

X : désignant les entrées, la matrice des observations
 m : étant le nombre capteurs, et le nombre de matrices d'intercovariance à diagonaliser
 p : spécifiant le nombre de matrices d'intercovariance à diagonaliser

Un calcul de l'estimée de la matrice de blanchiment en passant par la matrice de covariance

$\hat{\mathbf{R}}_x(0)$ en y faisant intervenir la puissance du bruit si besoin en est :

En effet, si le nombre de capteurs est différent de celui des sources,

Alors, la partie du signal \mathbf{R}_y sera déduite de $\hat{\mathbf{R}}_x(0)$ et donc la matrice de blanchiment sera

donc la racine carrée de \mathbf{R}_y .

Si, par contre le nombre de capteurs est pris égal au nombre de sources, la matrice cherchée est prise égale directement à la racine de la matrice $\hat{\mathbf{R}}_x(0)$.

Le blanchiment effectué, et les matrices d'intercovariances calculées, la diagonalisation conjointe est réalisée avec le calcul des matrices de rotations de Givens. Et par suite, la matrice source est estimée en multipliant le signal non blanchi avec l'inverse de la matrice du mélange préalablement estimée avec la matrice de covariance.

Plus concrètement ailleurs, SOBI a eu différentes applications et avec succès dans plusieurs techniques avancées. Dans cette suite, nous avons choisi cet exemple :

Exemple d'une application de SOBI : la magnétoencéphalographie (MEG)

L'un des terrains de prédilection de la séparation aveugle après les communications reste incontestablement le médical, et précisément, SOBI, à l'instar d'INFOMAX, ou de FICA (Hyvarinen et Oja, 1997), a eu son mot à dire dans ce domaine également.

SOBI a été appliquée avec succès aux signaux de l'électroencéphalographie (EEG) et de la magnétoencéphalographie. Des techniques de séparation pour extraire l'activité neuronale des fréquentes interférences :

- le clignement des yeux qui génère beaucoup de données indésirables)
- Le squelette de la boîte crânienne qui introduit des distorsions et affaiblit les faibles fréquences spatiales du champ électrique. Dans la magnétoencéphalographie, c'est un champ magnétique qui pénètre le squelette avec peu de distorsions. La précision de la localisation des sources neuronales est de l'ordre de quelques millimètres dans des conditions optimales,

ont subi des implémentations avec succès également.

Nous en citerons ceux de Makeig (1998), de Juig et al (1998), de Tang et al (1999), de Jung et al (1999), de Ziehe et al (2000), de Tang et al (2000)...

Comparativement, la nouveauté qu'a apporté SOBI, hormis l'exploitation des statistiques du second ordre, est que la résolution s'est faite sans modélisation, et, à l'image des techniques de séparation de sources, supprime ces interférences sans connaissances préalable des sources.

La conception expérimentale a été réalisée à Albuquerque, à l'Université du Nouveau Mexique (Etats-Unis) et s'est faite sur 4 sujets droitiers (2 hommes et 2 dames) leur faisant effectuer :

1. des tâches spécifiques impliquant un stimulus visuel à large angle qui activera une grande population neuronale distribuée sur une grande surface du cortex cérébrale. Le générateur d'une aussi grande surface ne peut être caractérisée par une source ponctuelle. Aussi des erreurs de localisations surviendront si c'est le cas, et de telles erreurs vont influencer

implicitement sur le l'estimation du temps de réponse. Alors que l'algorithme de la séparation de sources, SOBI ici, estime le temps de réponse sans faire d'hypothèse sur la position spatiale de la source et par conséquent, ne subit pas l'influence de la population activée, et donc ni sur l'étendue de la sonde activée.

2. des tâches qui induisent un stimulus auditif seulement pendant une durée assez court. La réponse correspondante neuronale peut changer selon le niveau d'attention. L'activation de la population auditive, sous cette atténuation rapide de l'attention est plutôt rare et difficilement modélisable en utilisant les méthodes conventionnelles.

3. des tâches impliquant le stimulus somatosensoriel comme quand le sujet appuie sur le bouton. Ici également, la modélisation serait assez médiocre pour localiser la source neuronale correspondante.

L'expérience a montré qu'en normalisant les vecteurs colonnes de W à 1 et en satisfaisant les conditions exigées, les composants somatosensoriels, visuels et auditifs sont séparés et en plus remis à l'échelle, sont identifiés à un dipôle équivalent qui est contenu dans un volume inférieur à 10mm^3 .

La conclusion des expérimentateurs s'énonce comme suit :

« On a analysé l'ensemble des données de 4 sujets, réalisant quatre tâches conçues à l'origine pour d'autres fins. Chaque tâche implique l'activation de stimulus somatosensoriel, visuel et auditif en plus de l'interaction avec le contexte naturel de l'activité cérébrale.

A travers l'application d'algorithme BSS(Blind Source Separation) aux données MEG, on a montré que BSS est capable de séparer différentes sources neuronales issues de différents processus, et que les composants séparés peuvent être localisés,..... »

Ils poursuivent :

« ...avec l'aide des algorithmes de BSS, il est possible de mesurer le temps de réponse des tests uniques sans entrer aucune sonde à l'intérieur du cerveau et avec une résolution de l'ordre du milliseconde....Les résultats nous suggèrent que la méthode de séparation de sources est robuste. L'application de SOBI a permis la détermination des composants issus du stimulus qui sont spatialement localisés dans les régions du cerveau où ils se manifestent. Les résultats obtenus concordent avec ceux déjà réalisés auparavant. Et la localisation des ECD(dipôle équivalent à de la source) tombe dans les régions où ils ont été déjà trouvés dans d'autres travaux. »

IV.3. Conclusion

Aujourd'hui, il est évident qu'au moment ces lignes se remplissent, des applications qui vont probablement révolutionner la science sont en train de s'appréhender et qui sait, sont déjà opérationnelles.

Séparation au second ordre: Algorithme SOBI

V.1. Introduction	23
V.2. Bases de la formulation du problème.....	25
V.3. Principe de base de l'algorithme SOBI.....	25
V.3.1. Les hypothèses	25
V.3.1.a. Les sources.....	25
V.3.1.b. Les perturbations.....	25
V.3.2. Développement	25
V.3.3. La normalisation du problème.....	25
V.3.4. La diagonalisation conjointe	25
V.3.4.a. Principe.....	25
V.3.4.b. Démonstration.....	25
V.3.4.c. Implantation de la diagonalisation conjointe.....	25
V.4. Conclusion.....	31

V.1. Introduction

Cette section tentera de répondre aux diverses interrogations suivantes :

Sur quoi se sont-ils appuyés ? Y a-t-il une ligne directrice ?

Comment se présente l'algorithme SOBI ? Quelles en sont les particularités ?

V.2. Bases de la formulation du problème

La formulation de notre problème repose sur des fonctions du traitement du signal dont la réalisation passera par l'élaboration de méthodes propres au traitement du signal. La mise en œuvre de ces méthodes s'envisage ensuite selon deux points de vue en général :

* Soit on considère que toutes les informations statistiques requises sont connues sans erreur et que l'on peut déterminer sans difficulté des moyennes d'ensemble, auquel cas la mise en œuvre en question débouche sur des *procédures*.

* Soit le point de vue s'avère délibérément non probabiliste et dans ces cas, seules les moyennes envisageables sont temporelles; la mise en œuvre de ces méthodes requiert alors *l'implémentation d'algorithmes* (en langage évolué, en assembleur, ...) où des problèmes d'une autre nature se posent telles que les erreurs d'arrondi, ou de troncature dont le calculateur est responsable, l'estimation sur un horizon d'observation fini des grandeurs caractéristiques intervenant dans les méthodes la manipulation de matrices, et plus généralement de données mal conditionnées,

Cette vision kaléidoscopique de la littérature en traitement du signal nous plonge d'ores et déjà dans le vif du sujet de la séparation de sources et conformément aux prochains chapitres, nous verrons que c'est le second point de vue qui nous intéressera principalement.

Et donc en conséquence, la séparation des sources peut engendrer différentes méthodes de résolutions selon la problématique liée directement à la disponibilité ou non de certaines informations. En effet, lorsqu'il y a corrélation, c'est-à-dire cohérence temporelle, une solution au problème de la séparation des sources peut être développée en exploitant que les statistiques d'ordre deux.

Et c'est dans ce sens qu'a été mis au point SOBI (Second Order Blind Identification), introduite par Bélouchrani et al. en 1995.

Dans un premier temps, notre première préoccupation sera de dresser les principales lignes directrices de cette technique avant d'examiner en profondeur l'algorithme sous des applications et implémentations.

V.3. Principes de base de l'algorithme SOBI

L'essentiel du travail repose sur la manipulation des observations.

Statistiquement parlant, les signaux sources seront séparés de par la manipulation des matrices d'intercovariance des signaux capteurs, manipulation qui repose sur la diagonalisation conjointe à travers des procédures de décomposition propres, possibles grâce à leur structure simple.

V.3.1. Les hypothèses

Elles s'énoncent comme suit :

V.3.1.a. Les sources (H1)

Elles doivent vérifier ces quatre points :

① La stationnarité au second ordre

$$\begin{aligned} \Rightarrow m_2(t_1, t_2) = E[s_i(t_1) s_j^*(t_2)] &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_i s_j^* p(s_i, s_j, t_1, t_2) ds_i ds_j \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s_i s_j^* p(s_i, s_j, t_2 - t_1) ds_i ds_j \end{aligned}$$

② La décorrélation mutuelle, ce qui implique une indépendance mutuelle c'est-à-dire. mathématiquement que

$\forall \tau$, pour $1 \leq i \neq j \leq n$, nous aurons : $E[s_i(t) s_j^*(t)] = 0$
ce qui permet d'écrire :

$$E[s_i(t) s_j^*(t - \tau)] = \rho_i(\tau) \delta_{ij} = R_s(\tau) \quad \forall \tau$$

Où δ_{ij} représente le symbole de Kronecker défini comme suit :

$$\delta_{ij} = \begin{cases} 1 & \text{pour } i=j \\ 0 & \text{pour } i \neq j \end{cases}$$

Donc R_s est une matrice diagonale dont les composantes sont les coefficients de covariance du processus $s_i(t)$

③ La moyenne à chacune est égale à zéro

$$\Rightarrow E[s_i(t)] = 0, \text{ et } \forall t, \forall i=1, \dots, n$$

④ Les puissances sont des valeurs finies

V.3.1.b. les perturbations (H2)

Le bruit est supposé être :

- ✓ Additif
- ✓ Stationnaire
- ✓ Temporellement blanc
- ✓ De moyenne nulle
- ✓ Et décorrélé des signaux sources

En effet, la supposition d'un bruit global blanc, gaussien, trouve son explication dans le théorème de la limite centrale qui affirme que si les hypothèses suivantes sont satisfaites :

1) les variables aléatoires x_1, x_2, \dots, x_n ont une distribution quelconque

- 2) leurs valeurs moyennes sont nulles ($E[x_1]=E[x_2]=\dots=E[x_n]=0$)
- 3) elles sont indépendantes ou leurs coefficients de corrélation tendent vers zéro
- 4) la valeur de chaque variable aléatoire individuelle est négligeable par rapport à la somme de toutes les variables

Alors la densité de probabilité de la somme

$$Y = \sum_{i=1}^n x_i \quad , \text{ pour } n \text{ très grand tend vers la loi normale, c'est-à-dire}$$

la gaussienne et

$$p(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y^2}{\sigma^2}\right)}$$

Concrètement, ceci répond au constat que même à partir de la douzième convolution de n'importe quels signaux, convolutions faites entre eux, nous obtenons une gaussienne. Ce qui peut justifier à notre niveau la modélisation de ce bruit dans la séparation.

Remarque : Cependant, nous ferons la remarque que SOBI résiste à n'importe quel bruit. Et donc nous pouvons écrire alors :

$$E [b(t) b^*(t-\tau)] = \delta R_b \quad (7)$$

De plus, en supposant que les composantes de la matrice bruit sont identiques, soit qu'elles sont de même puissance σ^2 , R_b s'écrira :

$$R_b(\tau) = \sigma^2 \delta(\tau) I, \quad (8)$$

où I est la matrice identité.

V.3.2 Développement

D'une manière générale, les matrices d'intercovariance des observations s'écrivent :

$$R_x(\tau) = E [x(t) x^*(t-\tau)] \quad (9)$$

(9) s'écrira d'après ce qui précède :

$$\begin{aligned} R_x(\tau) &= E [(As(t)+b(t)) (As(t-\tau)+b(t-\tau))^*] \\ &= E [(As(t)+b(t)) (s^*(t-\tau)A^*+b^*(t-\tau))] \\ &= E [(As(t)s^*(t-\tau)A^*+As(t)b^*(t-\tau)+b(t)s^*(t-\tau)A^*+b(t)b^*(t-\tau))] \\ &= E[As(t)s^*(t-\tau)A^*] + E[As(t)b^*(t-\tau)] + E[b(t)s^*(t-\tau)A^*] + E[b(t)b^*(t-\tau)] \\ &= \underbrace{AE[s(t)s^*(t-\tau)^*]A^*}_{R_s} + \underbrace{AE[s(t)b^*(t-\tau)]}_{R_{sb}} + \underbrace{E[b(t)s^*(t-\tau)]A^*}_{R_{bs}} + \underbrace{E[b(t)b^*(t-\tau)]}_{R_b} \\ &= A \quad R_s \quad A^* \quad + \quad \underbrace{0}_{R_{sb}} \quad + \quad \underbrace{0}_{R_{bs}} \quad A^* \quad + \quad R_b \end{aligned}$$

(décorrélation sources/bruit implique)

ce qui nous ramène à l'écriture générale suivante :

$$R_x(\tau) = \begin{cases} A R_s A^* + R_b & \text{pour } \tau = 0 \text{ (10a)} \\ A R_s A^* & \text{pour } \tau \neq 0 \text{ (10b)} \end{cases}$$

SOBI s'intéresse d'abord à l'équation (10b), lorsque le retard est non nul i.e

$\delta(\tau) = 0$, pour la connaissance de la puissance du bruit avant d'attaquer (10a) pour les matrices d'intercovariance.

V.3.3. La normalisation du problème

Le blanchiment de la partie $y(t)$ des observations se chargera de cette normalisation. L'objectif est de trouver une matrice dite de blanchiment W (de l'anglais Whiter), pour lequel appliqué au processus $As(t)$ on obtienne une décorrélation spatiale c'est-à-dire algébriquement on aurait :

$$E [(Wy(t))(Wy(t))^*] = WR_y W^H = WAA^H W^H = I_n$$

Autrement dit, WA serait une matrice carrée unitaire de dimension n

Une heureuse conclusion s'en déduit : Pour toute matrice de blanchiment W , il existe une matrice unitaire carrée liée par la relation :

$$WA = U \text{ ou encore } A = W^H U$$

La simplification introduite est immédiate :

Au lieu d'avoir affaire à déterminer une matrice de dimension $m \times n$, en l'occurrence la matrice mélange, la normalisation nous ramène à considérer uniquement une matrice unitaire U de dimension $n \times n$ plus réduite et de plus carrée. Des observations, d'un système de dimension $m \times n$, d'un nombre de sources de m , le blanchiment nous ramène à un système carré de dimension n et d'un nombre de sources égal à n .

Cette réduction est assez conséquente et de surcroît garde toute l'originalité des informations étant donné qu'un quelconque choix de la matrice unitaire laissera intacte l'autocovariance des observations blanchies.

Cet aspect de SOBI représente un enjeu potentiel essentiel quant à la résolution du problème de la séparation au second ordre. Comme illustration, il nous faudrait effectuer

$\{ m^2 n + m^2 (n-1) \}$ opérations élémentaires rien que pour une multiplication matricielle dans un système non blanchi au lieu de $\{ n^3 + n^2 (n-1) \}$ lorsque la normalisation sera faite. Si $m=5$ et $n=4$, par exemple:

Système blanchi \longrightarrow 112 opérations élémentaires
 Système non blanchi \longrightarrow 175 opérations élémentaires

On laissera le soin au lecteur de peser le pour et le contre, d'autant plus qu'on a été un peu trop raisonnable quant au choix de m et de n .

V.3.4 La diagonalisation conjointe

D'abord, partant du fait que les sources considérées comme stationnaires, centrées et de moyenne nulle de puissances finies et mutuellement indépendantes, la matrice $R_s(\tau)$ est diagonale avec à sa diagonale les puissances de tous les signaux sources.

Après blanchiment, les matrices d'intercovariance s'écriront à un retard différent de zéro :

$$\begin{aligned} \underline{R}_x(\tau) &= W R_y(\tau) W^H \\ &= W A R_s(\tau) A^H W^H \\ &= U R_s(\tau) U \end{aligned}$$

D'où l'on peut tirer que toutes les matrices d'intercovariances blanchies se diagonalisent tous sous la même matrice unitaire U .

Sous cette présente écriture, il est aisé de voir que les composantes de la matrice U ne sont autres que les vecteurs propres des matrices $\underline{R}_x(\tau)$ et la diagonale de $R_s(\tau)$ les valeurs propres associées pour un retard donné.

Et c'est de là qu'intervient la diagonalisation conjointe puisque tenter de résoudre le problème en ne considérant qu'une seule diagonalisation serait hasardeux car en présence de données réelles, tous les cas de figures peuvent se produire. Et dans le cas précis où les valeurs propres sont dégénérées c'est-à-dire, l'indépendance linéaire entre les vecteurs propres n'est pas observée, la résolution serait alors sérieusement compromise.

Cependant, vu que cette diagonalisation ne se réalise qu'avec des estimations et des valeurs empiriques, elle prendra plutôt le nom de diagonalisation conjointe approchée.

V.2.4.a. Principe

L'idée vient de cette remarque : « ...diagonaliser une matrice normale M de taille $n \times n$ est équivalent à trouver une base orthonormée $V_i = \{v_i / i = 1, \dots, n\}$ telle que dans cette base, la somme des modules carrés des termes non-diagonaux soit minimale... » [Bélou 1].

Autrement dit, il faudra minimiser :

$$C(V) = \sum_{i=1, i \neq j=1}^n \left\| [VMV^H]_{i,j} \right\|^2 \quad (11)$$

V.3.4.b. Démonstration

[Bélou 2]

Et vu qu'aussi la norme d'une matrice reste inchangée par une transformation unitaire et par la permutation (lorsque cela est possible)

$$\|VMV^H\|^2 = \|MVV^H\|^2 = \|MI_n\|^2 = \|M\|^2 \text{ or}$$

$$\begin{aligned} \|M\|^2 &= \sum_{i=1, j=1}^n \|[VMV^H]_{i,j}\|^2 \\ &= \text{le carré d'une matrice M quelconque} \\ &= \sum_{i=1}^n \|[VMV^H]_{i,i}\|^2 + \sum_{i=1, j \neq i=1}^n \|[VMV^H]_{i,j}\|^2 \end{aligned}$$

Donc minimiser (11) revient à trouver la base qui maximiserait :

$$C(V) = -\sum_i \left([VMV^H]_{ii} \right)^2$$

Et une diagonalisation conjointe d'un ensemble de matrices M_k est une extension du critère précédent, à savoir la minimisation de :

$$\begin{aligned} C(M, V) &= C(M_1, M_2, \dots, M_K, V) \\ &= \sum_{k=1}^K C(M_k, V) \\ &= \sum_{k=1}^K \sum_{i=1}^n \left([VMV^H]_{ii} \right)^2 \end{aligned}$$

soit en définitive:

$$C(V) = \sum_k C(M_k, V) = -\sum_{k,j} |v_i^* M_{kj} v_i|^2 \quad (12)$$

V.3.4.c. Implantation de la diagonalisation conjointe [Bélou 2]

La technique repose sur la généralisation de l'algorithme de Jacobi traitant de la diagonalisation d'une seule matrice hermitienne. Le critère sera minimisé progressivement en faisant appel à des rotations dites de Givens ; la matrice de Givens opérant sur une seule paire d'indices différents à chaque pas, i.e la diagonalisation ne concernera qu'une matrice carrée de dimension 2 et le processus sera répété jusqu'à passer en revue toutes les K matrices à diagonaliser.

Posons que ces matrices de Givens s'écrivent sous la forme générale de :

$$H_k = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix}$$

Position du problème :

On cherche une matrice unitaire V telle que les matrices H_k' ,

$$H_k' = V^H H_k V = \begin{bmatrix} a_k' & b_k' \\ c_k' & d_k' \end{bmatrix}$$

minimisent le critère (12).

La transformation unitaire V est paramétrée par une rotation de Givens complexe :

$$V = \begin{bmatrix} \cos\theta & e^{i\Phi}\sin\theta \\ -e^{-i\Phi}\sin\theta & \cos\theta \end{bmatrix}$$

θ et Φ étant alors les deux inconnues à trouver et qui minimiseront (12) soit qui maximiseront la somme $Q = \sum_k |a_k'|^2 + |b_k'|^2$

Avec :

$$|a_k' + d_k'|^2 + |a_k' - d_k'|^2 = 2(|a_k'|^2 + |d_k'|^2) \text{ et}$$

$$\text{Tr}(V^H H_k' V) = \text{Tr}(H_k' V V^H) = \text{Tr}(H_k' I_2) = \text{Tr}(H_k') = a_k' + d_k'$$

La maximisation se réduit alors :

$$Q = \sum_k |a_k' - d_k'|^2 \text{ avec}$$

$$a_k' - d_k' = (a_k - d_k)\cos 2\theta - (b_k + c_k)\sin 2\theta \cos \Phi - j(c_k - b_k)\sin 2\theta \sin \Phi$$

Le problème sera normalisé avec le choix de ces vecteurs :

$$\begin{aligned} u^T &= [a_1' - d_1', \dots, a_k' - d_k'], \\ v^T &= [\cos\theta, -\sin 2\theta \cos \Phi, \sin 2\theta \sin \Phi], \\ g_k^T &= [a_k - d_k, b_k + c_k, j(c_k - b_k)] \end{aligned}$$

On obtient K équations qui s'écrivent :

$$u = G v, \quad G^T = [g_1, g_2, \dots, g_k]$$

d'où

$$Q = u^H u = v^T G^H G v.$$

Et du fait que la matrice $G^H G$ soit hermitienne, et que sa partie imaginaire, l'écriture de la quantité se réduit définitivement à :

$$Q = v^T \text{Re}(G^H G) v; \text{ et la contrainte supplémentaire sur } v \text{ (} v^T v = 1 \text{)}.$$

La maximisation Q est obtenue en prenant ν comme le vecteur propre associé à la plus grande valeur propre de $Re(G^H G)$.

Remarque : Choix de «K»

Il est à noter que le choix de ce nombre est à peu près un compromis entre l'efficacité statistique et la lourdeur de nos calculs puisque l'obtention de la base sera d'autant plus précise que K est grand, et évidemment nous aurons beaucoup plus de matrices à diagonaliser.

En pratique, rien ne nous oblige à prendre ce nombre grand ou petit, ni avec quel type de variation il obéira ; nous prendrons en conformité avec [Bélou1], un pas régulier et unitaire et choisirons K comme étant *la longueur de corrélation* (durée assez suffisante pour que la fonction de corrélation soit significative) des signaux observés.

V.4. Conclusion

Le problème étant clairement identifié, nous passons directement à son solutionnement c'est-à-dire aux différentes étapes dictées par l'algorithme SOBI, que sont blanchiment, diagonalisation conjointe, estimation pour aboutir à la séparation en tant que telle.

Mais avant cela, nous nous permettrons d'introduire l'outil et le langage de travail à savoir respectivement l'XILINX et le VHDL

Chapitre 6

Le VHDL

VI.1. Introduction	31
VI.2. A propos du VHDL.....	31
VI.3. Pourquoi le langage de description VHDL?.....	31
VI.4. La conception VHDL.....	32
VI.5. Les niveaux d'abstraction.....	33
VI.5.1. Le niveau système.....	33
VI.5.2. Le niveau algorithmique.....	34
VI.5.3. Le niveau RTL.....	34
VI.5.4. Le niveau logique.....	35
VI.6. Conclusion.....	36

VI.1. Introduction

L'électronicien a toujours utilisé des outils de description pour représenter des structures logiques ou analogiques. Le schéma structurel que l'on utilise depuis si longtemps et si souvent n'est en fait qu'un outil de description graphique.

Aujourd'hui, l'électronique numérique est de plus en plus présente et tend bien souvent à remplacer les structures analogiques utilisées jusqu'à présent. L'explosion du marché de l'informatique, du téléphone cellulaire et des systèmes de communication de données en général a poussé les industries à développer des produits de plus en plus petit, consommant de moins en moins et offrant de plus en plus de fonctionnalités et de performance. Pour ce faire, ils ont créé des systèmes avec peu de Circuits Intégrés et des Circuits Imprimés de plus en plus petits. C'est pour répondre à cette situation que sont apparus les circuits logiques programmables (PLDs) et des langages de description hardware comme le VHDL (Very high speed integrated circuits Hardware Description Language)

Ainsi, l'ampleur des fonctions numériques à réaliser nous impose l'utilisation d'un autre outil de description. Il est en effet plus aisé de décrire un compteur ou un additionneur 64 bits en utilisant l'outil de description VHDL, par exemple, plutôt qu'un schéma.

VI.2 A propos du VHDL

Le Département de la Défense des U.S.A., dans une partie de son programme de recherche "Very-High-Speed Integrated Circuit"(VHSIC), a mis au point le VHSIC HDL (VHDL) en 1982. C'est une des quelques HDLs (Hardware Description Language) en général les plus utilisés aujourd'hui. Il a été reconnu comme langage de description hardware standard par l'Institut des Ingénieurs Electroniciens et Electriciens (IEEE Standard 1076, ratifié en 1987) et par le Département de la Défense des Etats Unis.

Révisée en 1993 pour supprimer quelques ambiguïtés et améliorer la portabilité du langage, cette norme est vite devenue un standard en matière d'outils de description de fonctions logiques. A ce jour, on utilise le langage VHDL pour :

- concevoir des ASIC,
- programmer des composants programmables du type PLD, CPLD et FPGA,
- concevoir des modèles de simulations numériques.

VI.3 Pourquoi le langage de description VHDL?

Sommairement, VHDL décrit le comportement, la fonction, les entrées et les sorties et les entrées /sorties d'un circuit digital. Le langage VHDL est comparable en matière de style et syntaxe aux langages modernes de programmations mais cependant inclut en plus des constructions spécifiques au hardware.

VHDL répartit les entités (composants, circuits, ou systèmes) en partie externe ou visible (nom de l'entité et connections) et en une partie interne ou cachée (algorithme et implémentation de l'entité). Après avoir défini l'interface externe en une entité, d'autres entités peuvent utiliser cette entité lorsqu'elles sont toutes en train d'être développées. Ce concept de vues interne et externe est fondamental et propre à la description VHDL d'un système.

En VHDL, des processus indépendants modèlent des circuits séquentiels utilisant des bascules, des latches, et des circuits combinatoires, en utilisant exclusivement des portes.

Un langage de description hardware comme le VHDL permet de décrire et de synthétiser rapidement des circuits contenant l'équivalent de 5 à 20 milles portes logiques. Un travail équivalent avec les équations Booléennes prendrait quelques mois de travail.

Le deuxième point fort du VHDL est d'être "un langage de description de haut niveau". D'autres types de langage de description, comme l'ABEL par exemple, ne possèdent pas cette appellation. En fait, un langage est dit de haut niveau lorsqu'il fait le plus possible abstraction de l'objet auquel ou pour lequel il est écrit. Dans le cas du langage VHDL, il n'est jamais fait référence au composant ou à la structure pour lesquels on l'utilise. Ainsi, il apparaît deux notions très importantes :

① La portabilité des descriptions VHDL, c'est-à-dire, possibilité de cibler une description VHDL dans le composant ou la structure que l'on souhaite en utilisant l'outil que l'on veut (en supposant, bien sûr, que la description en question puisse s'intégrer dans le composant choisi et que l'outil utilisé possède une entrée VHDL) ;

② La conception de haut niveau, c'est-à-dire qui ne suit plus la démarche descendante habituelle (du cahier des charges jusqu'à la réalisation et le calcul des structures finales) mais qui se "limite" à une description comportementale directement issue des spécifications techniques du produit que l'on souhaite obtenir.

Ailleurs, le VHDL comme tous les HDLs, dans les designs de systèmes est beaucoup plus pratique dans de nombreux projets où plusieurs designers ou groupe de designers sont en train de travailler en commun. Le VHDL fournit un développement structuré. En effet, après que les décisions principales d'architecture aient été prises et que les principaux composants et leurs connectives identifiées, le travail peut fonctionner indépendamment en sous-projets.

Une méthodologie de design qui utilise les HDLs possède plusieurs avantages fondamentaux relativement à la traditionnelle méthodologie niveau-porte. Certains de ces avantages sont :

- **Puissance et Flexibilité** : Les descriptions HDL fournissent une technologie indépendante de la documentation d'un design et de ses fonctionnalités. Il offre en plus la possibilité de créer des bibliothèques de composants réutilisables d'un projet à l'autre. Il permet aussi le design modulaire d'une application. On peut créer des composants qui seront eux-mêmes utilisés dans un composant plus complexe. L'utilité est grande car le code de chaque composant peut être validé par simulation; la recherche des erreurs dans un composant complexe est donc simplifiée. Une description HDL est beaucoup plus facilement lisible et compréhensible qu'une description schématique ou de la netlist.
- **L'indépendance** : Le VHDL permet de créer une fonction sans se préoccuper du composant dans laquelle elle sera implémentée. Il ne faut donc pas devenir intimement familier avec l'architecture d'un circuit particulier pour optimiser l'utilisation des ressources et les performances. Parce que la description initiale HDL d'un design est une technologie indépendante, on peut la réutiliser pour créer le design dans une technologie différente, sans avoir à la retraduire de la technologie originelle.
- **La portabilité** : Le VHDL permet de simuler le comportement d'un circuit. De part le caractère standard du langage, cette simulation peut être effectuée au moyen de n'importe quel simulateur. Le circuit peut être programmé par n'importe quel outil de synthèse et ce sur n'importe quel type d'ordinateur.

On peut vérifier la fonctionnalité assez tôt dans le processus et immédiatement simuler un design écrit dans une description HDL. La simulation du design à ce haut niveau, avant l'implémentation au niveau porte, nous permet de tester les décisions architecturales et conséquemment celles du design.

Un dernier argument de taille à ne pas oublier et justifiant la pleine utilisation du VHDL relève surtout du fait que les HDLs supportent principalement des descriptions à des niveaux mélangés, où les élaborations de la structure et de la netlist peuvent être assimilées à des descriptions comportementales ou algorithmiques. Avec cette capacité de mélange de niveau, on peut décrire des architectures de systèmes à un très haut niveau d'abstraction; Alors progressivement raffiner un design en une implémentation particulière niveau-composant ou niveau-porte. Alternativement, on peut lire une description HDL d'un design dans la *Fondation Express*, par exemple et alors diriger le compilateur pour synthétiser automatiquement l'implémentation niveau porte.

VI.4. La conception VHDL

Une fois que le cahier de charges est bien défini et la conception entamée, le designer se doit de respecter une ligne directrice tout au long de sa conception. Le processus de conception va du global au détaillé et suit des étapes bien ordonnées, de manière à concevoir l'ensemble de la manière la plus abstraite et la plus concise possible, et à reporter le plus loin possible le choix des détails d'implémentation, et surtout de respecter le plus possible cette indépendance des objets de façon à garantir cette particularité propre aux HDLs et en particulier au VHDL qui est de faciliter et/ou de rendre possible toute amélioration ou extension ou encore modification du système.

La conception répond à un certain nombre de règles et bien que le concepteur y glisse un peu de son originalité, la manière reste globalement la même.

VI.5. Les niveaux d'abstraction

Les niveaux en VHDL sont étroitement liés à la technologie à laquelle on fera appel. Cinq ont été définis : système, algorithmique, RTL, logique et physique.

VI.5.1 Le niveau système

Ce premier niveau peut être vu comme une vue assez globale de la réalisation.

A ce stade, ce sont des unités fonctionnelles qui stimulées par une ou plusieurs entrées présentent telles ou telles sorties et constituent de grands ensembles algorithmiques; cependant les détails de chaque exécution à savoir comment par exemple, pour notre cas, si on prenait SOBI comme témoin, comment le bloc (puisque à ce niveau le concepteur ne dévoile aucune tactique) de blanchiment opérerait-il concrètement ?

V.5.2 Le niveau algorithmique

Ce niveau d'abstraction à peu près semblable au niveau architectural est le premier maillon de la chaîne dans une vue générale dans un processus de conception.

Une description comportementale de toutes les structures de l'algorithme est à la base. Ce travail est d'abord textuel et ne répond qu'aux spécifications de départ c'est-à-dire structurelles. L'approche comportementale visant également à modéliser des composants Hardware justifie sa particularité des autres méthodes du seul fait qu'elle n'a pas besoin nécessairement de refléter de quelle façon la conception a été mise en application. C'est fondamentalement l'approche de la boîte noire. Elle modélise exactement ce qui se produit sur les entrées et les sorties de la boîte noire, mais ce qui est à l'intérieur de la boîte (comment cela fonctionne ?) est non pertinent.

La description comportementale est habituellement employée dans deux manières en VHDL.

D'abord, elle peut être employée pour modéliser des composants complexes qu'il serait pénible de décrire en utilisant les autres méthodes. Ceci pourrait être le cas par exemple, si nous souhaitions simuler dans notre opération de conception faite avec une la commande reliée à une partie comme un microprocesseur. Dans ce cas-ci, le microprocesseur serait complexe et son opération interne nous importe peu (seulement son comportement externe nous intéresse) ainsi elle serait probablement modélisée en utilisant le modèle comportemental.

En second lieu, les possibilités comportementales du VHDL peuvent être plus puissantes et sont plus commodes pour quelques conceptions. Dans ce cas-ci la description comportementale impliquera probablement une certaine structure de l'exécution.

Du point de vue structurelle, la synthèse algorithmique s'implémentera en faisant appel à des composants matériels tels les cartes et/ou les circuits intégrés par exemple. C'est au designer de spécifier selon son hiérarchisation comportementale si besoin est d'apporter quelques modifications répondant le plus souvent aux exigences d'espace ou de débit (dans les conceptions à plusieurs sous-algorithmes concurrentiels c'est-à-dire susceptibles de s'exécuter en même temps. Alors des décisions comme l'utilisation de tampons ou de l'augmentation d'unités physiques peuvent être prises. En effet toutes les transformations en vue de faciliter l'implémentation peuvent être acceptées. Une structure pipeline ou l'ajout de processus concurrentiels ou encore un découpage structurel peuvent alors surgir. Et progressant dans les modèles de représentation, le niveau algorithmique est vu dans le physique comme des systèmes exécutant les mêmes fonctions.

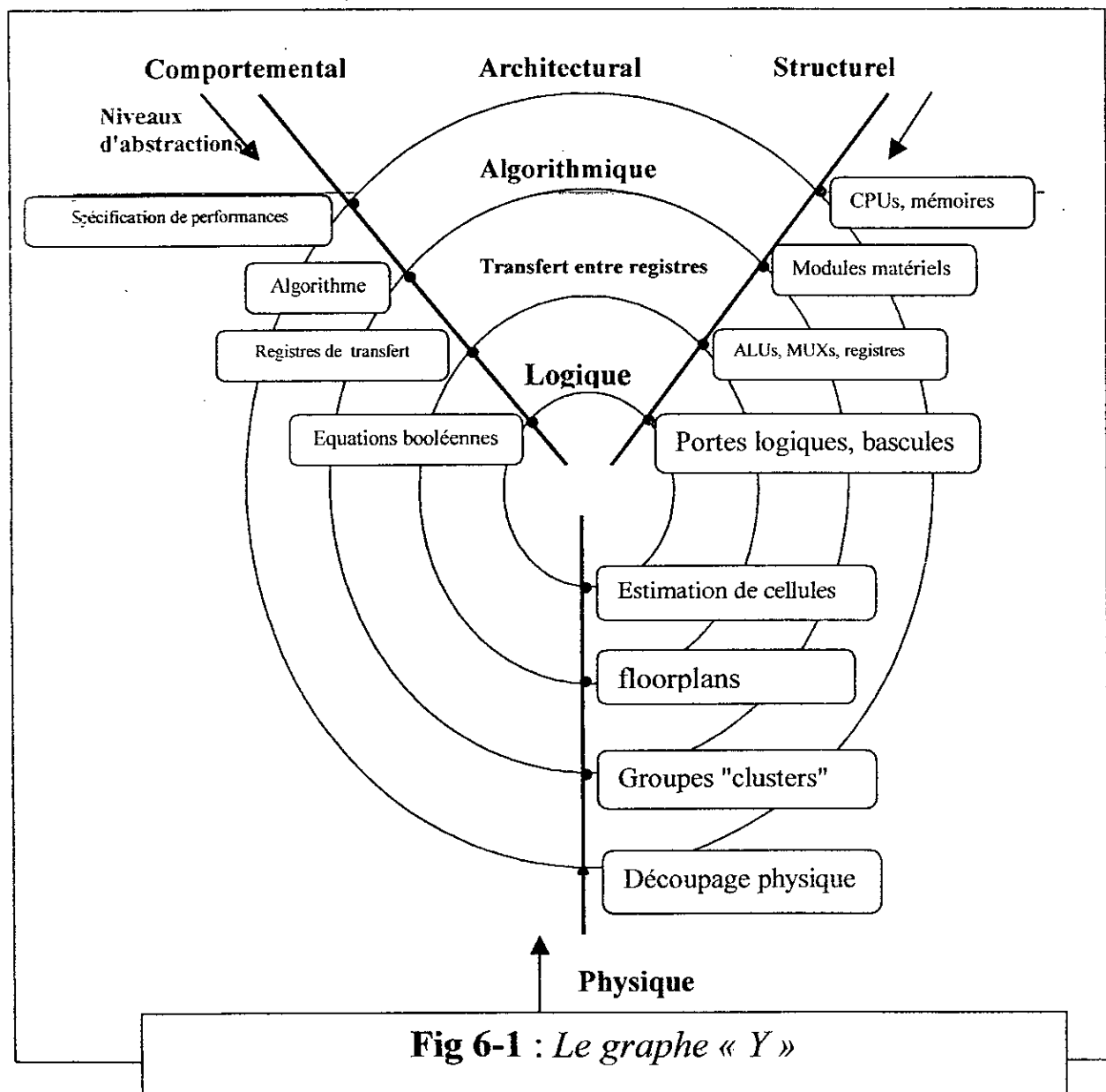
VI.5.3 Le niveau RTL (Register Transfert Level)

C'est une description qui est une traduction de la structure de départ au niveau logique et arithmétique. L'implémentation est faite fondamentalement avec des registres, des unités arithmétiques et logiques, des multiplexeurs, des mémoires ; le processus de conception est ramené à une opération de transfert de données. Et ceci implique bien évidemment tout un lot de spécifications à déterminer à savoir le nombre de registres et d'unités fonctionnelles à utiliser; Ces choix sous-tendent également toute la panoplie nécessaire pour l'intercommunication des différents modules passant par les variables et fonctions, les données, les mémoires, ... Ici également, l'aspect concurrentiel du VHDL aura un très grand effet quant à l'ordonnement des instructions et la plupart du temps, le concepteur doit faire appel immanquablement à l'éditeur de contraintes.

VI.5.4 Le niveau logique

Plongeant encore de plus en plus, le niveau qui pouvait bien prévaloir après les registres n'est autre que le niveau logique. On peut considérer ce niveau comme notre « kernel ». En effet,

comportementalement analysé, il correspond à une constellation d'équations booléennes ne s'implémentant alors qu'avec des portes logiques. Notons cependant que la synthèse logique offre certains avantages à savoir que l'on aboutit à une conception qui sera très flexible aux modifications ultérieures, et les documents générés permettent une plus rapide exploitation. A titre illustratif, nous vous proposons ce diagramme pour résumer cette première étape.



VI.6. Conclusion

Des concepts nouveaux se sont développés dans le langage et bien que les règles de programmation en VHDL soit figées, la technique et les styles de développement restent carrément très personnalisés. Et grâce à cette puissance offerte, le VHDL est en passe de devenir le langage de description hardware le plus utilisé dans le monde bien que nouvellement venu.

Chapitre 7

Développement en VHDL

VII.1. Introduction.....	37
VII.2. Estimation du blanchisseur.....	38
VII.2.1 Estimation de la matrice de covariance.....	38
VII.2.2 Diagonalisation.....	38
VII.2.2.a. Détermination des valeurs propres de $\hat{R}(0)$	39
VII.2.2.b. Détermination de vecteurs propres.....	43
VII.2.2.c. Estimation de la puissance du bruit.....	45
VII.2.2.d. La matrice de blanchement \hat{W}	45
VII.3. Blanchiment des données.....	45
VII.4. Estimation des K matrices d'intercovariances $\hat{R}(k)$ de $\hat{x}(t)$	46...
VII.5. Diagonalisation conjointe approchée des matrices d'intercovariance	46..
VII.6. Estimation de la matrice de mélange.....	48
VII.7. Estimation des signaux sources.....	49
VII.8. Conclusion.....	50

VII.1. Introduction

En première vue, nous allons introduire l'algorithme tel que présenté dans la référence [1] et nous servira de support tout au long de la programmation.

Algorithme SOBI (Second Order Blind Identification)

- ① Estimation du blanchisseur
- ② Blanchiment des données : $\hat{\underline{x}}(t) = \hat{W} x(t) ; \quad t=1,2,\dots,T$
- ③ Estimation des K matrices d'intercovariances $\hat{R}(\tau_k)$ de $\hat{\underline{x}}(t)$
à différents retards $\tau_k, \quad k=1,2,\dots,K$
- ④ Diagonalisation conjointe approchée des matrices d'intercovariances
dans une base $\hat{U} = [\hat{u}_1, \dots, \hat{u}_n]$
- ⑤ Estimation de la matrice de mélange par : $\hat{A} = \hat{W}^{\#} \hat{U}$
- ⑥ Estimation des signaux sources par : $\hat{\underline{s}}(t) = \hat{A}^H \hat{R}(0)^{-1} \hat{\underline{x}}(t)$

Fig 7-1 : Algorithme SOBI

VII.2 Estimation du blanchisseur

Comme précédemment souligné, cette étape nécessitera la détermination des valeurs propres de la matrice de covariance R_y . Et R_y sera obtenue à partir de R_x , matrice de covariance des observations mais à un retard nul, conformément à la relation (10a).

VII.2.1 Estimation de la matrice de covariance

Ce travail sera fait après avoir recueilli T échantillons .

$$\hat{R}_x(0) = \frac{1}{T} \sum_{t=1}^T x(t)x^*(t)$$

Implantation

Nous sommes en présence de données à valeurs complexes, aussi il nous faut impérativement dans un premier lieu définir ce type dans notre programme avant d'espérer pouvoir le manipuler puisqu'il est non défini aussi bien dans le VHDL'87 que dans le VHDL'93.

Ce passage réussi, nous pouvons prétendre au calcul de la matrice de covariance qui nous intéresse.

Les observations de départ, qui constituent en fait nos valeurs de base et qu'on utilisera pratiquement dans la quasi-totalité du travail, seront sauvegardées dans une ROM et seront appelées au besoin.

VII.2.2 Diagonalisation

La diagonalisation de cette matrice fournira cette écriture :

$$\hat{R}_x(0) = H\Lambda H^T$$

où les vecteurs colonnes h_i , $i = 1, \dots, n$ sont les vecteurs propres associés aux valeurs propres λ_i de l'estimée de la covariance. La matrice regroupe ces valeurs propres dans sa diagonale.

Implantation

Cette étape est des plus essentielles; tout d'abord le choix de la méthode ne se pose plus: première chose, déterminer les valeurs propres de la matrice, qui ne sera faite qu'après le calcul du polynôme caractéristique, et de sa résolution, ensuite nous servir de cela pour calculer les vecteurs propres associés et enfin et seulement, la diagonalisation sera effective.

VII.2.2.a Détermination des valeurs propres de $\hat{R}(0)$

Cette étape est principale pour parfaire la diagonalisation des observations estimées. Pour cela, plusieurs méthodes numériques s'offrent à nous ; Dans ce travail, nous nous proposons la méthode dite de LEVERRIER, qui nous semble la plus optimale et la plus adéquate, et qui consiste à déterminer les coefficients du polynôme caractéristique de notre matrice dont on approchera ses racines c'est-à-dire ses valeurs propres par une autre méthode de résolution numérique : la méthode de GRAEFFE.

> Le polynôme caractéristique : Méthode de LEVERRIER

Les coefficients du polynôme sont donnés par la formule :

$$\begin{cases} a_k = -\frac{1}{(k+1)} [a_{k-1}S_1 + \dots + a_0S_{k-2} + a_1S_{k-1}] \\ P(x) = a_1x^m + a_2x^{m-1} + \dots + a_{m+1} \quad , a_1 \neq 0 \end{cases}$$

Nous avons x_1, \dots, x_m , les racines de ce polynôme et le polynôme caractéristique de $\hat{R}(0)$ est

$$P(\lambda) = \det(A - \lambda I) = (-1)^m (\lambda^m - \sum_{k=1}^m a_k \lambda^{m-k}) \quad \text{d'ou } S_k = \text{tr}(A^k) \quad \text{et } a_1 = (-1)^m$$

estimons le nombre d'opérations élémentaires nécessaires à la détermination des

$$\{a_k\}_{k=1, \dots, m}$$

- Pour calculer S_k , il faut un produit de matrice et calculer une $\sum_{i=1}^m$, c'est à dire

$$\left. \begin{array}{l} \cdot m^2 m \text{ multiplications} \\ \cdot m^2 (m-1) \text{ addition} \\ \cdot m (m-1) \text{ additions} \end{array} \right\} \text{pour le produit des matrices pour calculer la trace.}$$

Pour calculer a_k , il faut

k multiplications

$k-1$ addition et

une division

donc pour déterminer les a_k , $k=2, \dots, m+1$, il faut

$$m^4 + \sum_{k=1}^m k = m^4 + \frac{m(m+1)}{2} \text{ multiplications}$$

$$m^3(m-1) + \sum_{k=1}^m (k-1) = m^4 - m + \frac{m(m+1)}{2} \text{ additions}$$

m divisions

soit un total de $T_L = 2m^4 + m^2$ opérations élémentaires pour $m=5$, obtient donc $T_L = 1275$ et pour $m=10$ $T_L = 20100$.

> L'Algorithme

Les arguments supplémentaires sont les suivants :

C : désigne un tableau de travail à une dimension qui contiendra, les coefficients du polynôme caractéristique.

B : désigne un tableau de travail à 2 deux dimensions qui sert à calculer les puissances successives de R.

S : désigne la matrice à une dimension qui sert à stocker les traces S_k .

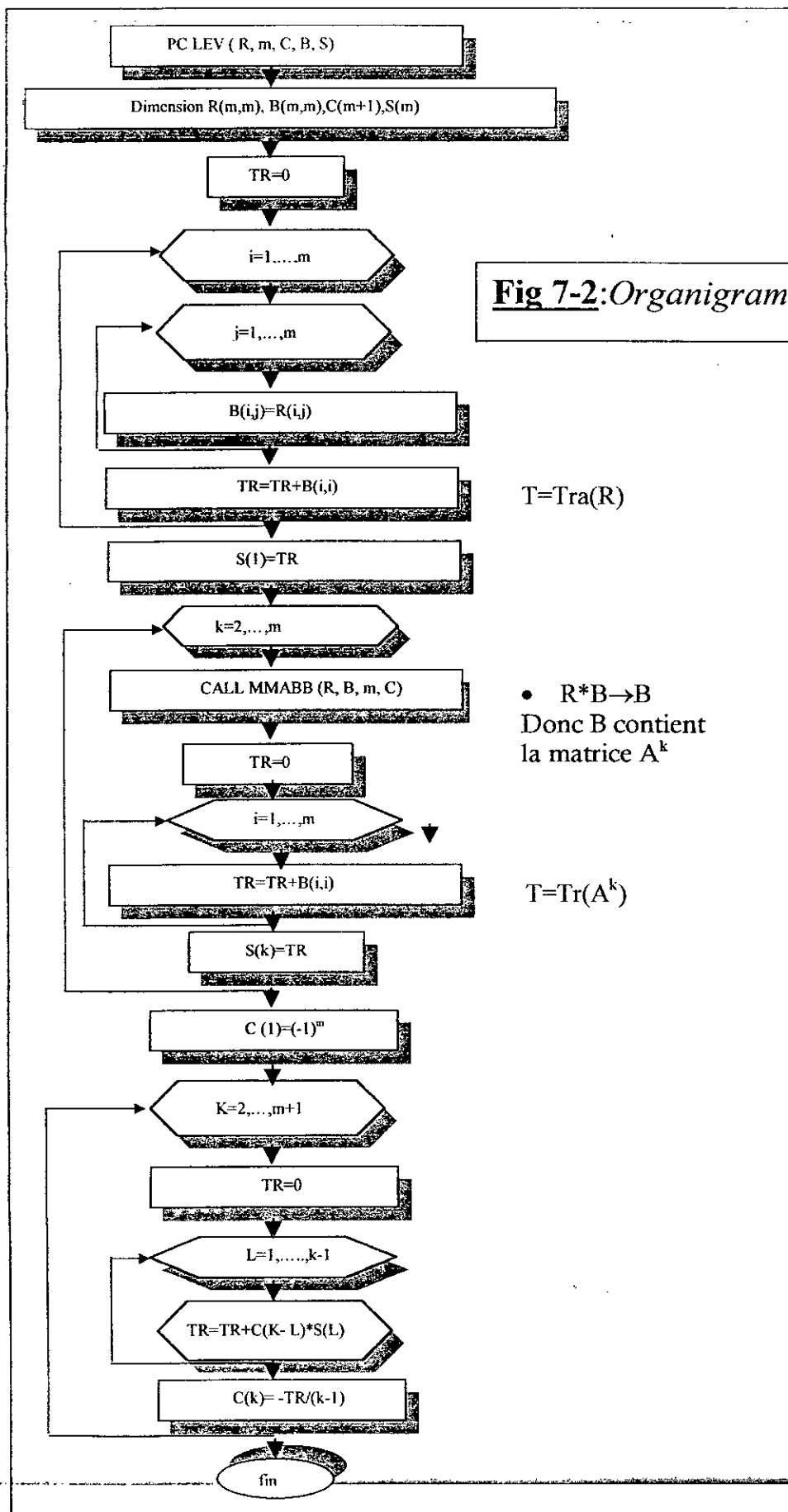
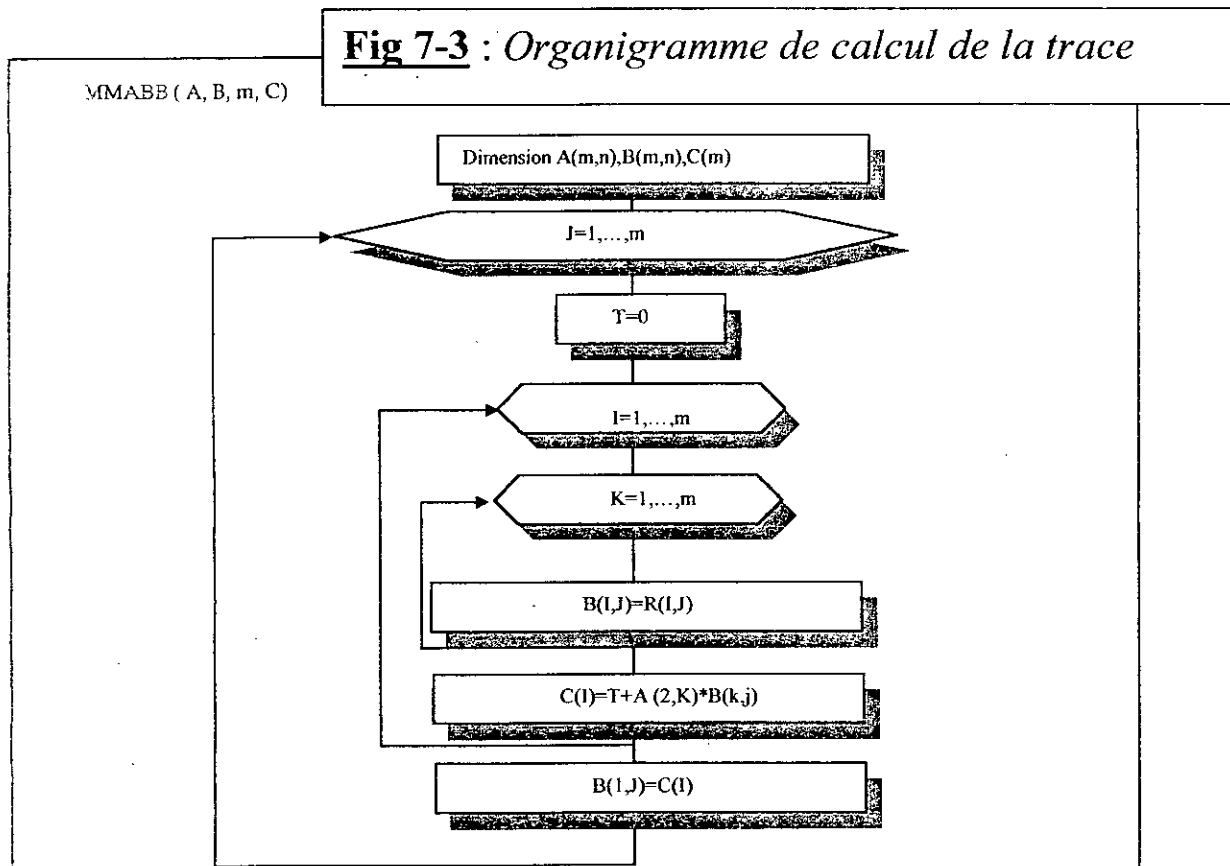


Fig 7-2: Organigramme de Leverrier

$T = \text{Tra}(R)$

• $R * B \rightarrow B$
Donc B contient la matrice A^k

$T = \text{Tr}(A^k)$



➤ Résolution de l'équation algébrique : Méthode de GRAEFFE

L'algorithme de Graeffe se chargera de nous donner les valeurs propres que l'on cherche.

Présentation de la méthode :

la méthode de GRAEFFE qui, dans le calcul de P_{2^k} , élève certains coefficients à la puissance 2^k risque d'aboutir rapidement à un test de dépassement de capacité en ordinateur. L'organigramme prend deux précautions :

1) Les coefficients du polynôme caractéristique sont normalisés, i.e. divisés par $A(1)$. on donc $B(1) = C(1) = 1$.

On se donne un majorant CM du tableau C inférieur au nombre réel machine maximum toléré.

Si l'une des composantes du tableau C n'est pas contenu dans l'ensemble

$$E = CM, -/CM] \cup \{0\} \cup [1/CM, CM]$$

On arrête les itérations.

2) la méthode de GRAEFFE ne donne pas le signe de la racine x_0 . pour lever cette ambiguïté, on calcule $P_n(x_0)$ et $P(-x_0)$ par le schéma d'Horner et choisit alors le signe qui donne la valeur absolue minimale.

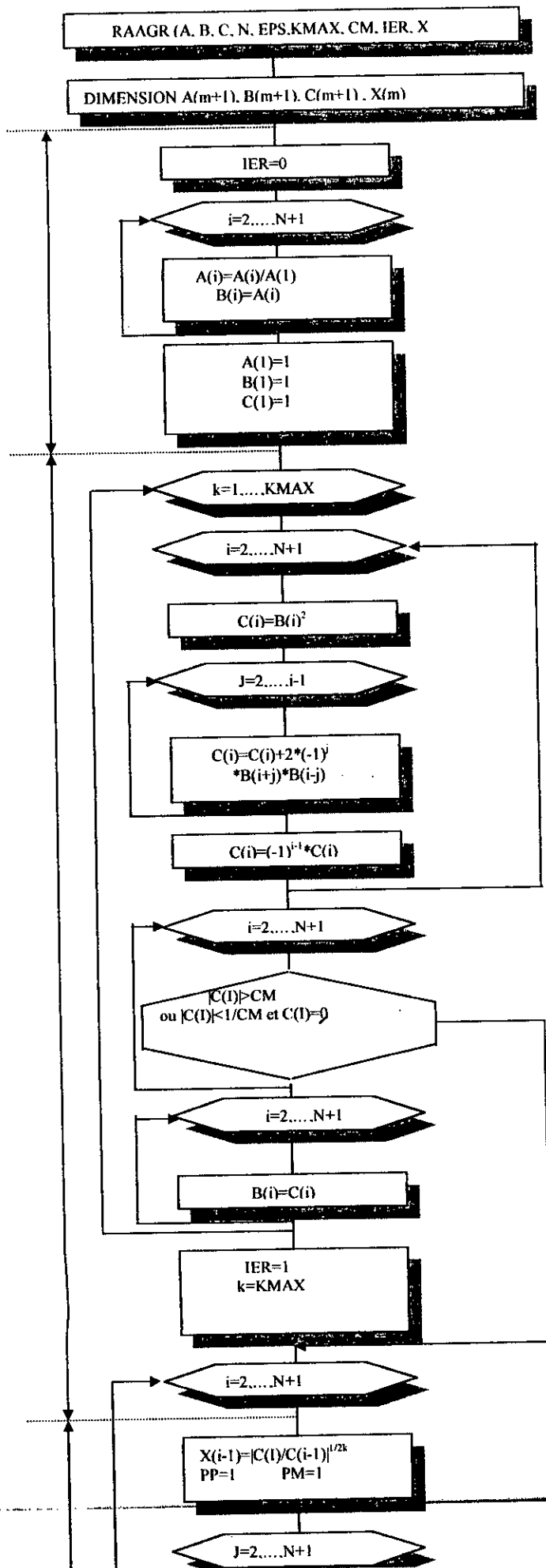
Les arguments de cette algorithmme sont les suivants :

A : désigne un vecteur de dimension $(m+1) \times 1$ contenant les coefficients du polynôme caractéristique $P(x)$

B : désigne un vecteur de dimension $(m+1) \times 1$ contenant les coefficients du polynôme $P(x^2)$

C : désigne un vecteur de dimension $(m+1) \times 1$ contenant les coefficients du polynôme

$$P(x^{2^{(k+1)}})$$



ATTENTION : I+J ne doit pas dépasser N+1

Ce test est seulement destiné rester dans les intervalles de définition numérique des nombres réels propres à l'ordinateur sur le quel on travaille

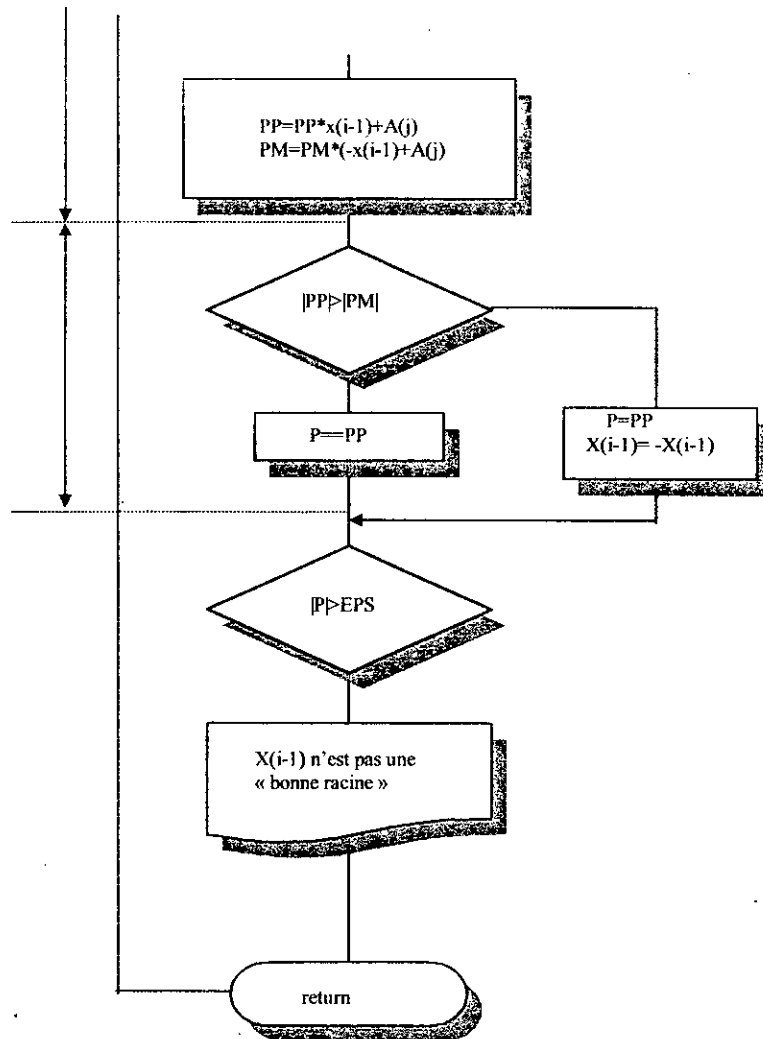


Fig 7-4 : Organigramme de Graeffe

A la fin de ce programme, nous aurons enfin les valeurs propres.

VII.2.2.b Détermination de vecteurs propres

Les valeurs propres étant connues la détermination des vecteurs propres associés est conséquente ; et pour cela, nous userons de l'algorithme de Trylov.

La formule est :

$$v_i = \frac{1}{d(i)} [x_{n-1}^2 + \beta_{i1}x_{n-2} + \dots + \beta_{i,n-1}x_0] , \begin{cases} d(i) = (-1)^m \frac{P(\lambda_i)}{\lambda - \lambda_i} \\ i = 1, \dots, m \end{cases} \quad \text{et} \quad \begin{cases} \beta_{i0} = 1 \\ \beta_{ij} = \lambda_i \beta_{ij-1} - a_j \end{cases}$$

$$x_0 = \alpha_1 v_1 + \dots + \alpha_m v_m$$

$$(I) \begin{cases} x_1 = Ax_0 \\ Av_1 = \lambda_1 v_1 \end{cases} \Rightarrow x_2 = A^2 x_0 ; x_3 = A^3 x_0$$

$$x_1 = \lambda_1 \alpha_1 v_1 + \dots + \lambda_m \alpha_m v_m$$

$$x_2 = \lambda_1 \alpha_1 v_1 + \alpha_2$$

avec v_i étant le vecteur propre d'ordre i

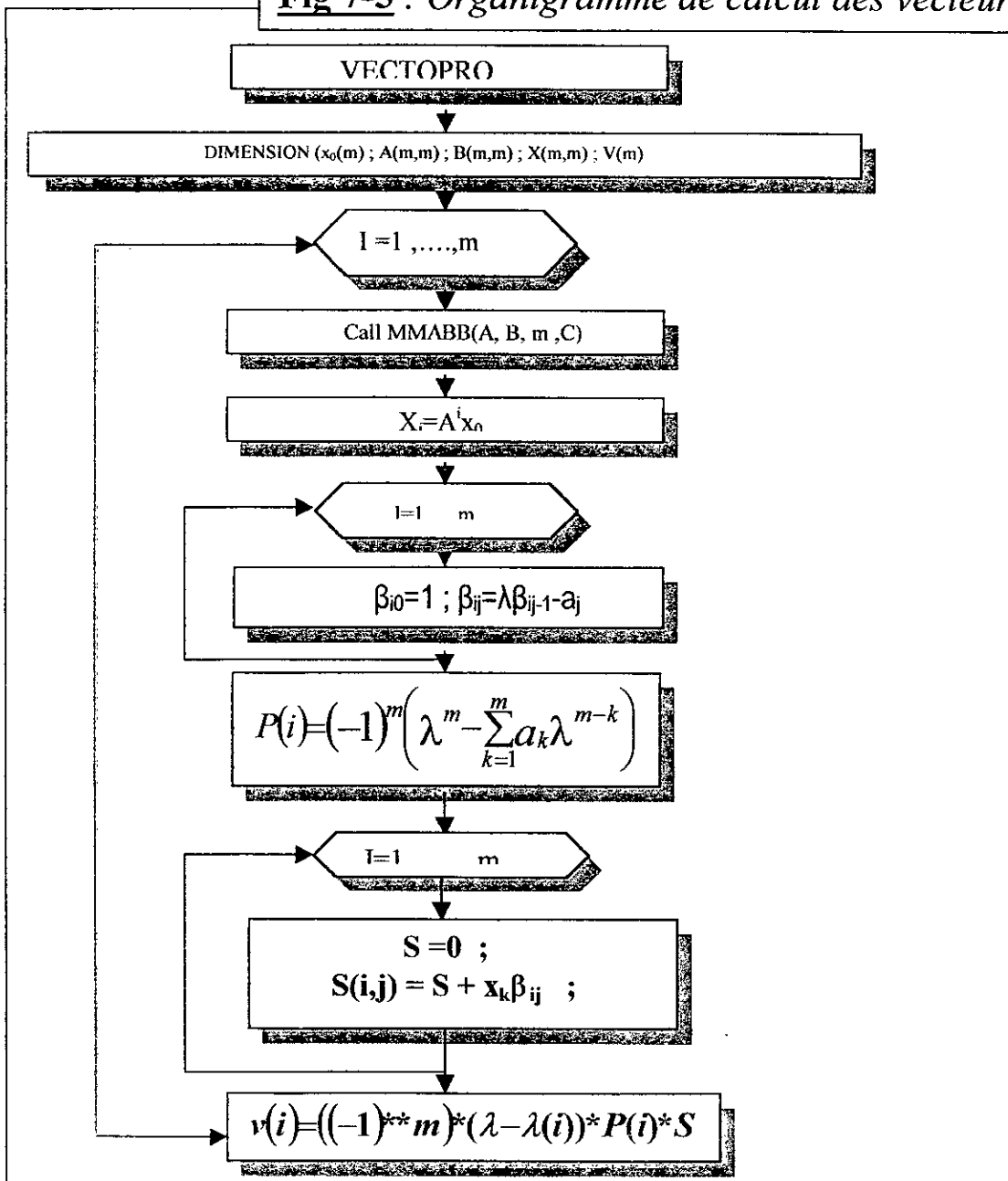
λ_i étant la valeur propre associée

β_j étant des coefficients

α_j étant des scalaires

a_j étant les coefficients du polynôme caractéristique

Fig 7-5 : Organigramme de calcul des vecteurs propres



Aussi, à la suite de ce travail, la matrice $\hat{R}(0)$ est entièrement diagonalisée.
Prochaine étape : Calcul de la puissance du bruit

VII.2.2.c Estimation de la puissance du bruit

Tout d'abord, il nous faudra ordonner les valeurs propres obtenues dans la partie (b) dans un ordre croissant et extraire les $(m-n)$ plus petites valeurs demandées pour le calcul de la puissance suivante :

$$\hat{\sigma}^2 = \frac{1}{m-n} \sum_{i=n+1}^m \lambda_i$$

Cependant, il faudra au préalable ordonner les valeurs propres trouvées, stade indispensable dans la détermination de la puissance.

VII.2.2.d La matrice de blanchement \hat{W}

$$\hat{W} = H \Delta^H = \left[\left(\lambda_1 - \hat{\sigma}^2 \right)^{-1/2} h_1, \dots, \left(\lambda_n - \hat{\sigma}^2 \right)^{-1/2} h_n \right]^H$$

$$H = (h_1, h_2, \dots, h_n) \leftarrow n \text{ au lieu de } m$$

$$\Delta = \text{diag} \left[\left(\lambda_1 - \hat{\sigma}^2 \right)^{-1/2}, \dots, \left(\lambda_n - \hat{\sigma}^2 \right)^{-1/2} \right]$$

Tous les paramètres cités ci-dessus sont connus et la description VHDL est alors immédiate.

L'étape ② peut maintenant être engagée.

VII.3. Blanchiment des données

Nous ferons appel encore à nos observations sauvegardées dans la ROM pour cette fois-ci les blanchir. Le travail qui sera effectué ne sera qu'un appel de valeurs et une multiplication matricielle concernant principalement la matrice trouvée au ① avec nos observations

$$\hat{x}(t) = \hat{W} x(t); \quad t=1,2,\dots,T$$

VII.4. Estimation des K matrices d'intercovariances

$\hat{R}(\tau)$ de $\hat{x}(t)$

L'estimation est déduite de la formule suivante :

$$\hat{R}_x(\tau) = \frac{1}{T-\tau} \sum_{t=\tau+1}^T \hat{x}(t) \hat{x}^*(t-\tau) \text{ avec}$$

$$\tau \in \{\tau_k, k=1, 2, \dots, K\}$$

Ici également on a affaire à une multiplication matricielle.

VII.5. Diagonalisation conjointe approchée des matrices d'intercovariances

Dans cette étape, l'objectif est de trouver une base V pour laquelle la forme quadratique Q est maximale ; et conséquemment au chapitre (iii), ceci se réalisera en prenant v comme le vecteur associé à la plus grande valeur propre de la matrice $Re(G^H G)$.

Dans une première partie, nous allons déterminer les valeurs et les vecteurs propres associés avant de statuer. Et pour cela, nous allons faire appel à la même procédure introduite pour la

diagonalisation simple de $\hat{R}(0)$. Ensuite, nous allons ordonner ces valeurs propres dans un même vecteur et passer à l'extraction de la plus grande pour lui associer son vecteur propre. Et comme v est fonction de θ et de ϕ , la transformation unitaire V est alors entièrement définie.

Mais avant cela, le passage par le calcul du vecteur G s'impose d'abord. Il est en effet composé d'une série de 3 vecteurs colonnes d'une matrice de Givens. Aussi, nous allons le définir pour les besoins du calcul comme un seul vecteur colonne de $3K$ composantes.

Le schéma que nous nous proposons est le suivant :

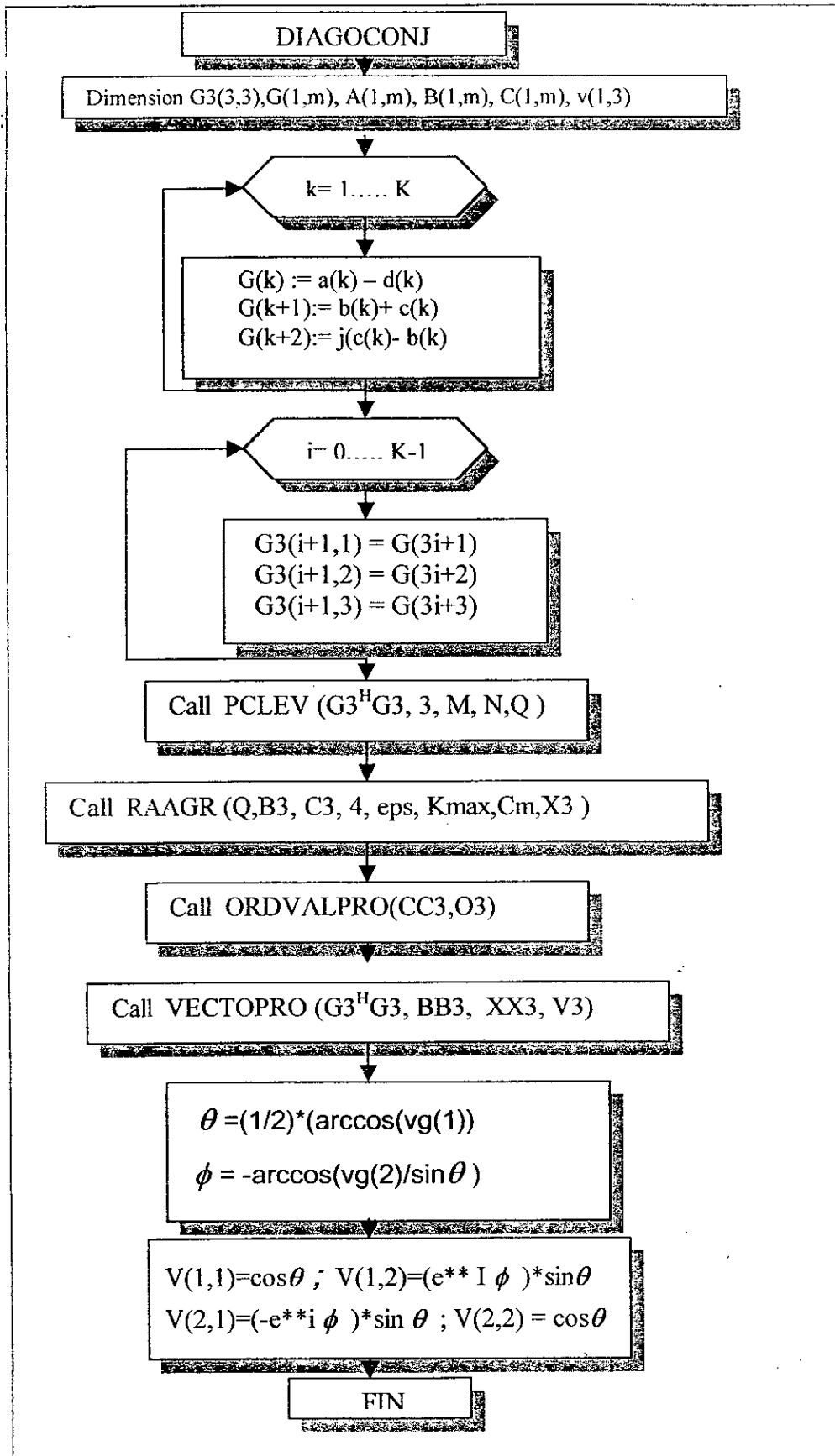
Les composantes a_k , b_k , c_k , et d_k seront chargés quatre vecteurs de longueur K .

$$A \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{pmatrix} \quad B \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{pmatrix} \quad C \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{pmatrix} \quad D \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_K \end{pmatrix}$$

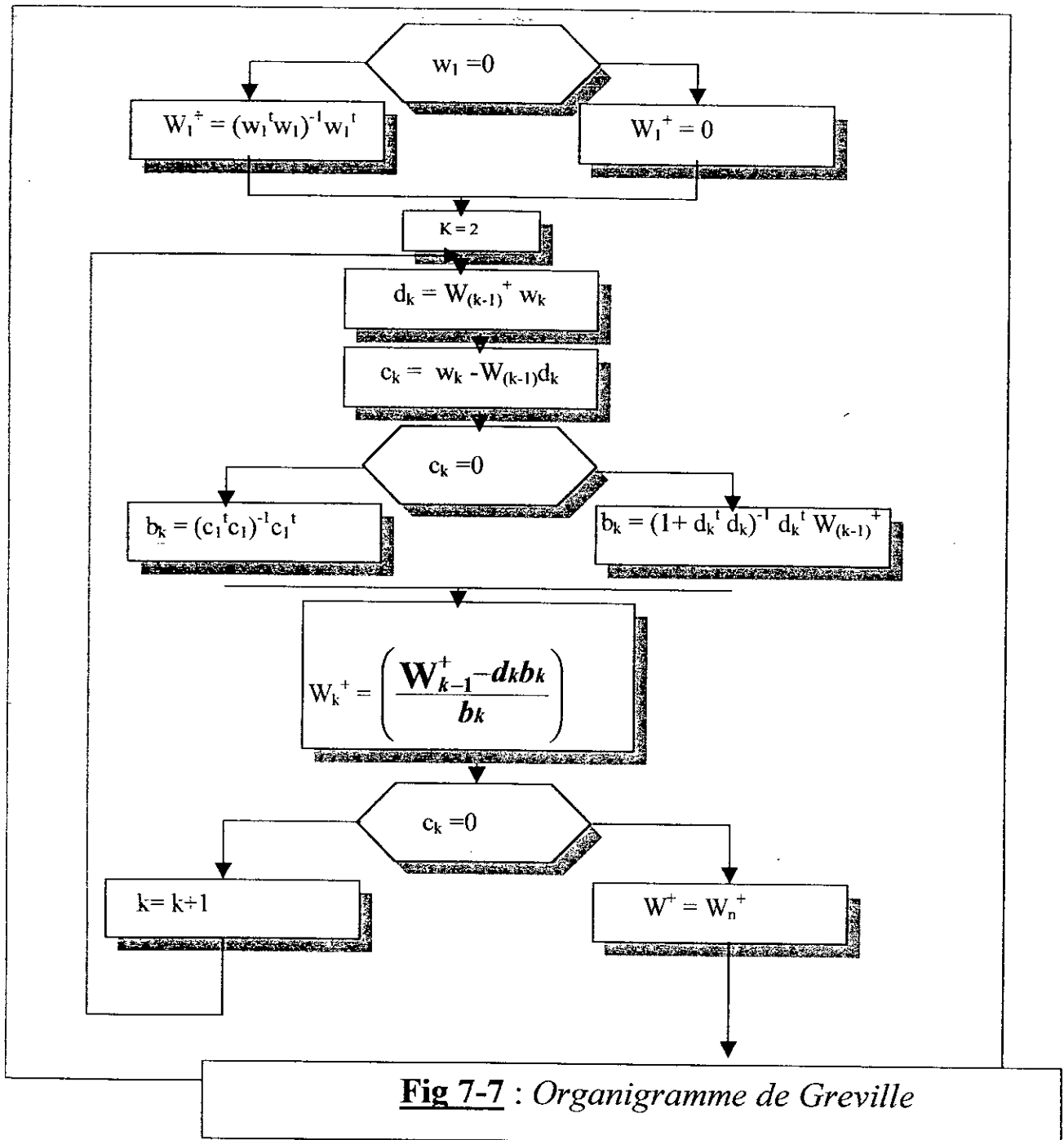
Et le vecteur G s'écrira alors :

$$G \begin{pmatrix} G_1 = a_1 - d_1 \\ G_2 = b_1 + c_1 \\ G_3 = j(c_1 - b_1) \\ \vdots \\ G_{3K} = j(c_K - b_K) \end{pmatrix}$$

Ensuite, cette écriture vectorielle sera transformée par une autre construction de telle sorte à avoir une matrice hermitienne pour $G^H G$. C'est-à-dire, ce vecteur G sera réorganisée de telle sorte à avoir des éléments matriciels (3,3). Et seulement après, le calcul de $G^H G$ peut s'effectuer pour en déduire sa partie réelle. L'algorithme se présente alors comme suit :

**Fig 7-6** : Organigramme de la diagonalisation conjointe

➤ Algorithme de GREVILLE



La matrice du mélange est alors immédiate et la multiplication matricielle fera l'affaire.
Et la dernière partie peut être lancée à savoir :

VII.8. Estimation des signaux sources

La formule utilisée sera : $s(t) = \hat{A}^H \hat{R}(0)^{-1} x(t)$

VII.9. Conclusion

Le développement que nous nous proposons de réaliser dans notre mémoire sera assez complexe. Les programmes correspondants seront écrits bien entendus en VHDL. Et ceci constituera la base de la conception, seulement, il sera question également de faire appel à toutes les ressources aussi bien logicielles, logiques, mathématiques, voire matérielles qui nous sera offertes avec nos outils de synthèse pour arriver à nos fins.

Le langage sera présente pour environ quatre vingt-dix pour cent dans la partie descriptive de l'algorithme, sans cependant être exclusif puisque, besoin nous y obligeant, notre projet de conception aura à intégrer d'autres ressources d'XILINX

VIII.1. Introduction

Le sursaut informatique allié aux besoins continuels de la miniaturisation dans un environnement électronique où technologies et innovations amplifient complexité et fonctionnalité, ne pouvaient guère laisser indifférent le monde des designers.

Des styles de conception de circuits ASIC furent alors mis sur place: la technologie Full-Custom, la technologie Standard Cells, la technologie Gate Array et enfin la technologie des composants programmables.

A chaque famille est associée un certain niveau de complexité, un type et une méthodologie de conception spécifiques.

VIII.2. Les technologies de conception

VIII.2.1. Les circuits sur mesure (Full Custom)

Ce sont les premiers appareils. Leurs spécificités :Chaque élément actif est fait sur mesure avec comme objectif premier l'optimisation de ses performances électriques et chaque bloc conçu doit avoir une surface utile de silicium minimale. Ils se distinguent de leur coût assez élevé et sont particulièrement présents dans les développements justifiant de densité d'intégration très grande (mémoire, capteurs d'images,...) ou de performances élevées (électronique rapide, électronique de puissance, analogique, faible bruit, ...).

VIII.2.2. Les circuits précaractérisés (Standard Cells)

Les circuits précaractérisés sont des circuits conçus à partir des objets informatiques délivrés par le fondeur de silicium. Spécificités : circuits de performances élevées, densité d'intégration accrue, mise en œuvre avec la CAO.

VIII.2.3. Les circuits prédiffusés

Ce sont des réseaux constitués de matrices de portes ou de cellules plus ou moins élaborés déjà diffusés dans le silicium mais non interconnectés. Ce sont des produits semi-finis alors qui nécessitent des masques. Spécificités : coût faible, cycle de conception court.

VIII.2.4. Les composants programmables

Leur prolifération est incontestable de nos jours et constituent les véritables fers de lance de l'électronique modernes. C'est la famille des PLD(PROM, PAL, PLA) qui sont des composants complètement achevés et personnalisables par l'utilisateur à l'aide d'un outil de développement, et des FPGA. Spécificités : conception, fabrication et maintenance simples de composants complexes, modifiabilité, rapidité (circuits temps réels) .

VIII.3. Les FPGA

On les définit comme étant des PLD complexes. Un PLD n'est rien d'autre qu'un réseau de portes AND connecté à une matrice de portes OR. Ils deviennent des PAL si la partie ET est seulement programmable, des PLA si les deux parties sont programmables.

Un FPGA allie à la fois un PLD et ce qu'on nomme un MPGA 'Mask Programmable Gate Array' qui est un circuit VLSI dans lequel toutes les couches qui définissent la circuiterie de la puce sont prédéfinies par le fabricant, sauf les couches métalliques qui permettent de réaliser les connections entre les transistors de la matrice MPGA.

Avec cette nouveauté, une très grande densité d'intégration est ainsi assurée.

Les premiers FPGA ont été introduits par l'XILINX en 1985, ensuite suivèrent des compagnies telles que Actel, Altera, Advanced Microdevices (AMD), Quick Logic,.....

Architecture d'un FPGA d'XILINX

Elle est composée de trois parties :

- un périmètre d'entrées/sorties : les IOB (Inputs/Outputs Blocks) qui coïncide avec l'interface programmable,
- une matrice noyau : les CLB (Configurables Logic Blocks) qui réalise la fonction personnalisées par l'utilisateur,

Le CLB constitue la cellule de base du FPGA. Elle se compose de deux générateurs de fonction de 16 bits (F ou G), un générateur de fonction de 8 bits (H), deux registres (des bascules ou des verrous), et des commandes reprogrammables de routage (multiplexeurs).

- et les ressources d'interconnexions qui servent de transport pour les signaux.

Les figures suivantes montrent à titre d'illustration les structures internes d'un CLB et d'un IOB de la famille XC4000.

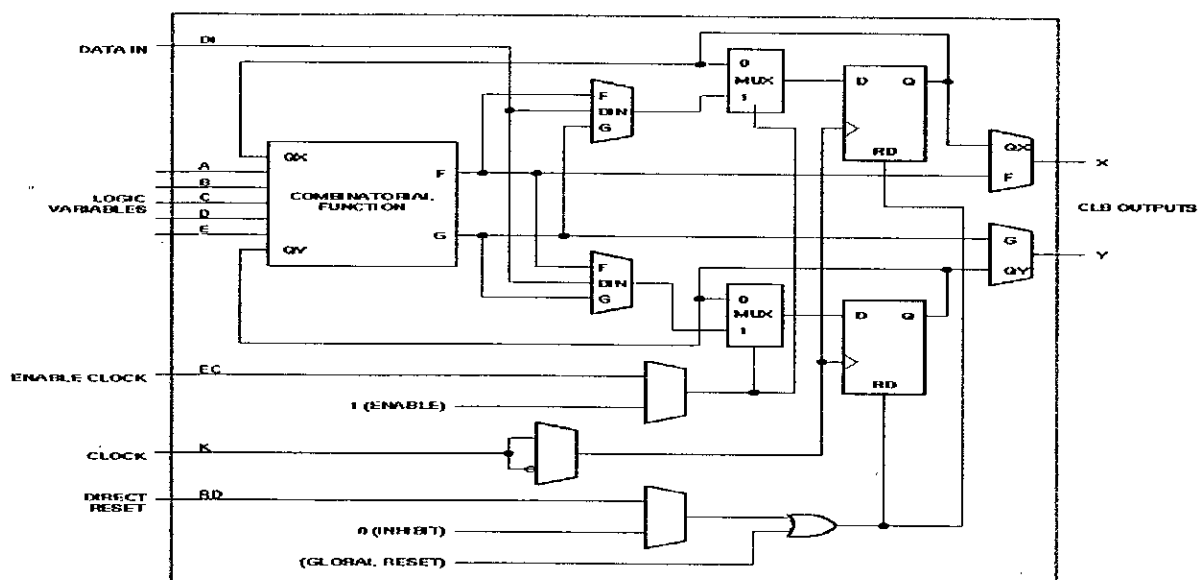


Fig 8-1 : Configuration interne d'un CLB (XC4000)

X3032

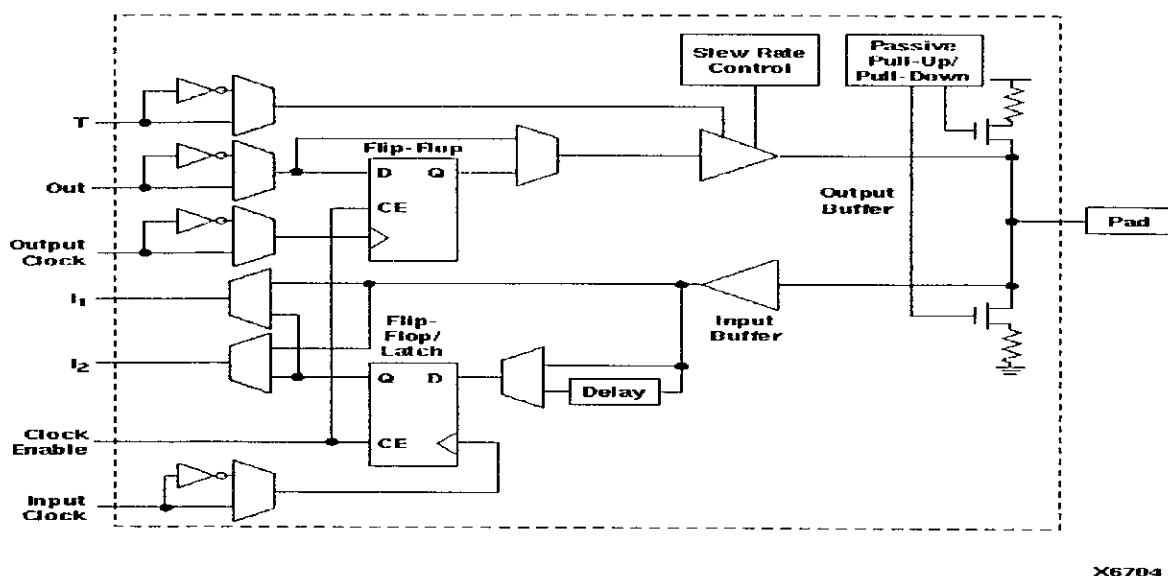


Fig 8-2 : Configuration interne d'un IOB (XC4000)

Généralement un circuit FPGA est caractérisé par deux catégories de blocs logiques :

1. les structures basés sur les tables de transfert (Look-up table based) : les LUT-based,
2. les structures basées sur les multiplexeurs (multiplexor-based) : les MUX-based.

Une LUT peut implémenter n'importe quelle fonction booléenne de m entrées ($m \geq 2$) d'où son nom de générateur de fonctions.

Les ressources d'interconnexions sont dominées par des interconnexions reprogrammables ou « one time programmable » ie programmable une seule fois. Et pour implémenter un élément programmant, nous avons toute la panoplie des RAM, SDRAM, les transistors EPROM,... L'XILINX utilise la programmation SDRAM.

VIII.4. La conception avec l'XILINX

Cette conception obéit au schéma de la figure 8-3 et peut se résumer comme suit :

1. Tout d'abord, la description est faite soit avec les langages hardware(format texte), soit à l'aide de schémas (symboles graphiques) ou avec les deux.

2. Ensuite, c'est la phase de l'optimisation c'est-à-dire la traduction de l'entrée précédente dans une autre forme qui pourra être plutard implémentée efficacement. Une première simulation peut être alors lancée à la fin de cette étape.

La troisième phase est la plus critique, c'est l'implémentation qui débute avec le mapping, qui est une représentation de toute la logique de la conception en ressources XILINX, en d'autres

termes le mapping attribue la logique en une autre compréhensible par l'XILINX, et cette nouvelle logique n'est autre que les IOB et les CLB.

Le schéma suivant est un récapitulatif des fichiers (les plus essentiels) générés lors d'une conception

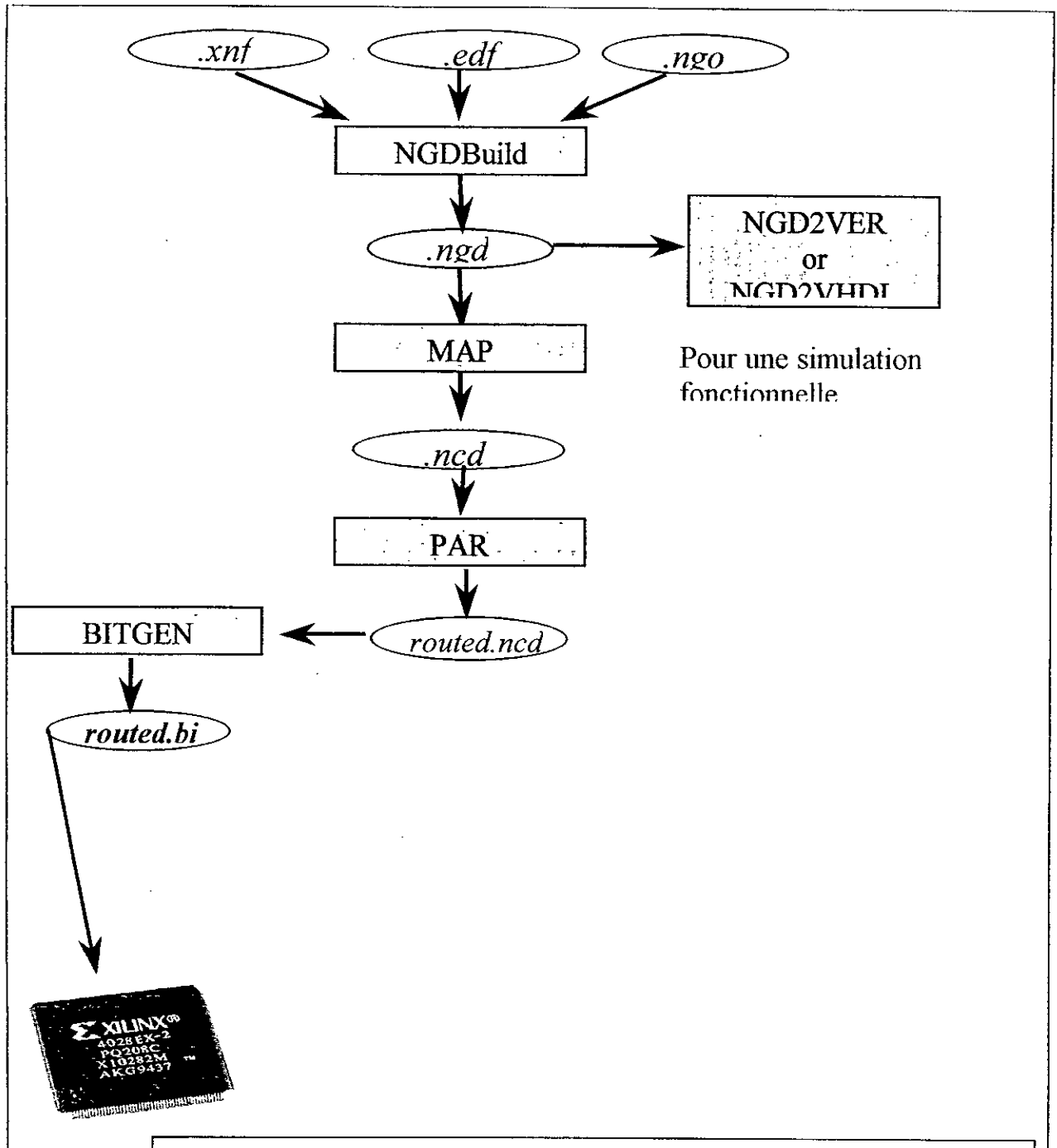


Fig 8-3 : La synthèse avec l'XILINX

↳ La quatrième étape est le placement qui parfaire une configuration spatiale des blocs créés lors du mapping.

↳ La cinquième est le routage de tout ceci.

Il y a création du fichier netlist décrivant les interconnexions entre les éléments logiques du composant.

↳ La sixième est la génération d'un bitstream, flot de données résumant la logique pour un composant FPGA. Il définit entièrement l'objet, c'est-à-dire, l'ensemble des opérations effectuées, localisation des IOB et des CLB, des connexions, ... toutes les informations y sont. Le fichier ainsi créé possède une extension .BIT.

↳ La back-annotation est la septième et joue le rôle d'analyse des performances entre le circuit original et celui qui vient d'être implémenté.

↳ La dernière concerne la vérification qui se à travers fera une simulation temporelle.

VIII.5. Conclusion

Le marché des FPGA a carrément inondé le milieu électronique en lui offrant un nouveau visage qui ne cesse de se perfectionner encore et encore.

Dans notre application, nous allons faire appel à cette nouvelle technologie qui est des plus usitées dans le contexte actuel. Toutes les étapes de la conception seront passées en revue, du logiciel au matériel.

Seulement, il convient de noter que par manque de dispositifs nécessaires pour une vérification hardware de notre travail, des parties non moins essentielles seront omises dont essentiellement l'implémentation réelle sur FPGA.

Chapitre 0.

Les résultats

IX.1. Introduction.....	56
IX.2. Le management du projet VHDL.....	56
IX.2.1. L'écriture des conditions.....	57
IX.2.2. Notre "top level"	57
IX.2.3. La Conception Détaillée.....	59
IX.2.4. Exécution.....	59
IX.2.5. Vérification par composants.....	59
IX.2.6. Intégration.....	60
IX.3. Le code de SOBI en VHDL.....	60
IX.3.1. Composition générale d'une description VHDL.....	60
IX.3.2. Les outils de développement.....	62
IX.4. Analyse des résultats.....	62
IX.4.1. La synthèse.....	63
IX.4.2. La simulation fonctionnelle	64
IX.4.3. L'implémentation.....	65
IX.5. Les performances..	67
IX.6. Conclusion.....	69

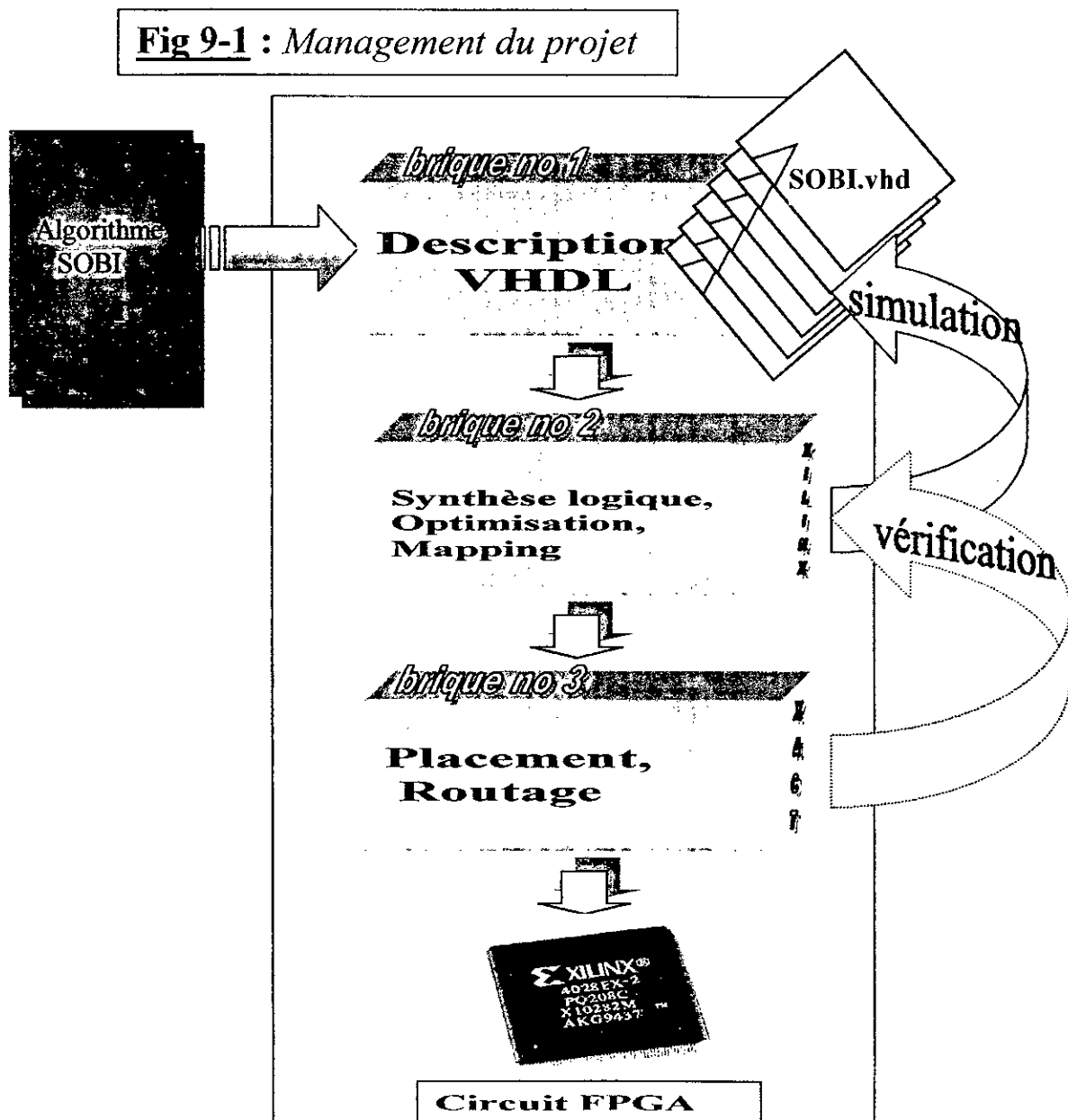


IX.1. Introduction

L'approche de conception est l'approche descendante et sera basée principalement sur la synthèse avec le VHDL. Une architecture est fixée, suivie d'un mapping de bas niveau pour permettre une simulation et par suite d'une implémentation et l'on clora cette partie avec une analyse des performances.

IX.2. Le management du projet VHDL

La conception suivra cette ligne directrice traduite par ce schéma ci-dessous :



IX.2.1 L'écriture des conditions

Le développement des conditions est une étape très critique en général dans une conception basée VHDL. A titre indicatif, les études de développement au début des années 80 ont indiqué que plus de 90 pour cent des réalisations ont passé avec succès l'étape de la simulation alors que moins de 50 pour cent seulement fonctionnaient dans le système une fois installées. Ces échecs sont le résultat du système fournissant un stimulus qui n'a pas été prévu par la conception ou le système s'attendant à une réponse qui n'a pas été incorporée. C'est un résultat des conditions mal définies. Cependant, l'exactitude et la perfection des conditions ne peuvent pas être vérifiées jusqu'à ce qu'après le dispositif soit fabriquée et installée dans le système. Toutes les erreurs dans les conditions auront comme conséquence des mises à jour de conception, des répétitions de tous les essais de vérification, et une différente fabrication à la fin. Ce processus est cher. Les erreurs de conditions sont faciles à faire, difficile à vérifier, et causent des résultats désastreux.. Cette tâche devrait être accomplie par les concepteurs ou les experts aînés.

En ce qui nous concerne, nous pouvons dire que cette étape nous a été prédéfinie au début de notre travail: l'algorithme doit être décrit en VHDL et simulé. Ce qui nous ramène à l'écriture de deux sortes de programmes, une qui satisfait aux exigences imprimées par le sujet et dictées par le promoteur et un autre que l'on se propose de rédiger pour avoir une implémentation.

Aussi, la tâche ne peut être facile, d'autant plus que SOBI, telle qu'il nous a été présenté traite, comme nous l'avons dit en plus haut, *des valeurs complexes*; ceci ne nous dérangera en rien dans la première étape de la conception, c'est-à-dire la description. La création de nouvelles bibliothèques définissant ce type résoudra ce problème mais bien partiellement. En effet, bien que le type réel soit reconnu, sa synthèse demeure pour le moment impossible et pour raison principale: L'IEEE ne l'a pas encore défini comme standard. Le logiciel XILINX ne traduira pas ce type, encore moins le type complexe. Les outils de synthèse n'étant pas incorporés dans le logiciel.

Les programmes dont les résultats sont présentés dans les *annexes B* et *C* ne peuvent alors correspondre à ceux satisfaisant aux exigences premières.

Cependant, nous avons décidé d'adopter une seconde méthode de conception tout en essayant au plus de donner une dimension beaucoup plus « circuit » à notre travail plutôt que de nous arrêter à une simple description, c'est-à-dire que nous avons décidé d'adopter un style de programmation pour enfin atteindre non seulement la simulation mais également l'implémentation de la totalité de nos programmes.

Aussi a-t-on défini cette ligne directrice.

IX.2.2. Notre "top level"

La conception supérieure dite « top level » a la plus grande influence sur tout le reste du programme. Nous démarrerons notre travail avec une description VHDL et appellerons au besoin tout autre composant à la manière d'un objet informatique à partir de la conception de base. A la question de savoir pourquoi est-ce que notre top level est un fichier VHDL, nous serions tentés de répondre que ce choix répondait aux exigences de notre thème qui est une description VHDL de SOBI tout d'abord, cependant il est bien évident que ce travail serait aussi respecté si le top level était un schéma, ou peut-être même un diagramme d'état; Notre

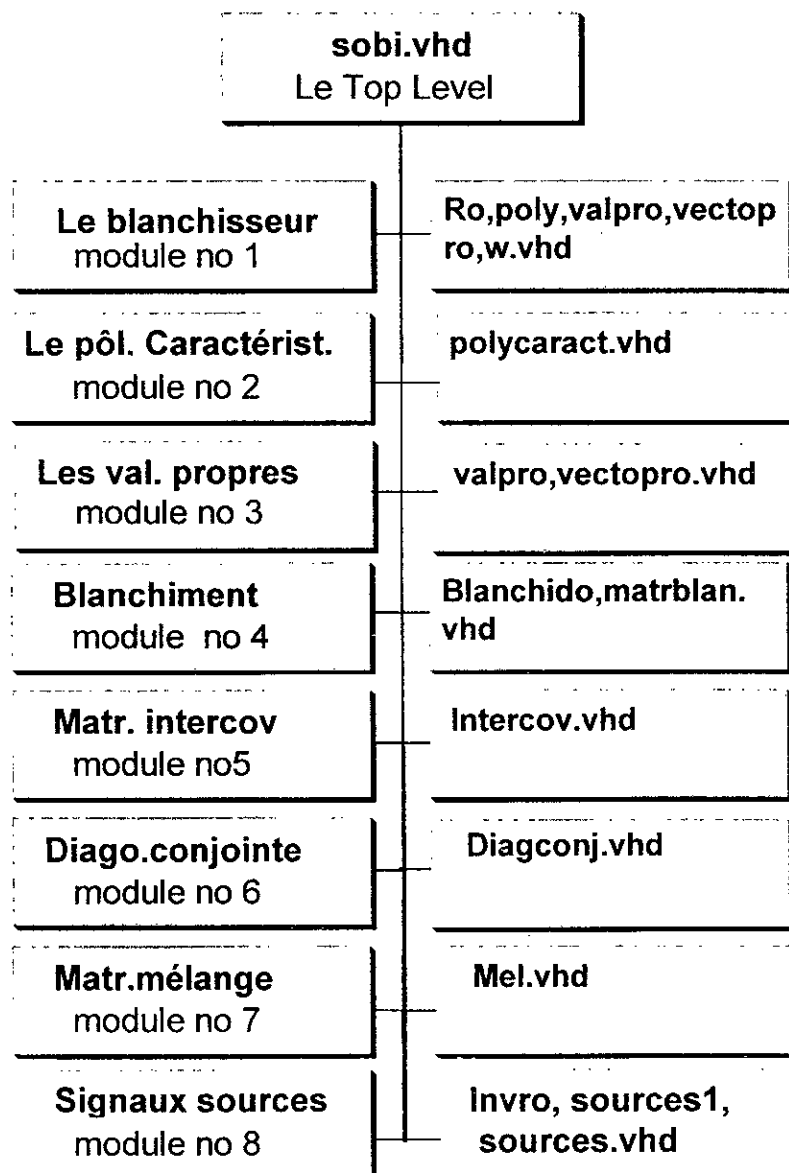
seule justification relève de la difficulté ou de la facilité relative d'appeler d'autres composants à travers le langage plutôt qu'avec n'importe quel autre outil de conception. Pour ne pas nous étaler, nous dirons qu'il est beaucoup plus facile de travailler avec le VHDL.

Les nouveaux concepteurs que nous sommes nous imposent de continuellement passer en revue les objectifs de la conception et les caractéristiques de la technologie dont on dispose avec notre logiciel de travail. Une conception de la sorte réduira au minimum l'interconnexion parmi les composants supérieurs.

Dans une implémentation, la vérification de la conception n'est pas complète jusqu'à ce que l'intégration de dispositif soit complète. Ceci exige répéter plusieurs d'essais de vérification. Même un changement mineur de l'interface entre deux composants peut avoir comme conséquence une glissade d'un mois de programme, en raison de la re-synthèse et de la re-vérification.

Le travail a été subdivisé en plusieurs sous-projets; dans chaque sous-projet, nous aurons affaire à des descriptions partielles, des simulations les validant tout de suite après, pour ne pas avoir à resimuler globalement et à recompiler tout le projet. Et pour chaque entité, nous présenterons l'implémentation et les résultats sous forme de circuits réalisés à l'aide de XILINX.

Aussi, nous avons défini une stratégie de conception divisée en six parties qui sont les suivantes :



- ① Estimation du blanchisseur
- ② Le blanchiment des données
- ③ La détermination des matrices d'intercovariance
- ④ La diagonalisation conjointe
- ⑤ La matrice du mélange
- ⑥ Les signaux sources

IX.2.3. La Conception Détaillée

La conception détaillée revient en quelque sorte à exécuter la conception supérieure. Cependant, l'impact du cheminement et de la complexité globale des entités est moins critique à ce niveau.

Nous optons pour une division d'un composant en composants atomiques. Ce choix sera justifié ultérieurement. Nous appellerons un composant atomique un composant qui sera mis en application comme seule paire d'entité-architecture et exhaustivement examiné après chaque changement. Un composant atomique peut être un élément spécifique de projet ou ce peut être un composant tout usage existant déjà dans une bibliothèque ou prévu pour être inclus dans une bibliothèque. On définit des composants (objets) qui peuvent être réutilisés par d'autres projets ou futurs projets. Un objet atomique a deux caractéristiques importantes: il est facile de mettre en application dans une architecture classée raisonnable et il est suffisamment indépendant qu'il puisse être exhaustivement examiné après chaque modification.

En fait, vu que nous ne disposons que de la version étudiante et de « maigres » bibliothèques pour l'implémentation de nos programmes, cette forme de conception sera notre seul salut pour prétendre à une implémentation avec des commentaires apparaissant dans le *map report* du genre « the design is too large for the specific device..... » ou carrément pour éviter d'avoir à rédiger des programmes kilométriques aussi incertains que non synthétisables.

Et spécialement, ces composants seront puisés principalement dans le logicore, le logiblox et même le « schematic »

IX.2.4. Exécution

L'exécution dans ce contexte se rapporte au code de l'écriture VHDL par la synthèse. La synthèse est la conversion finale du code de VHDL dans un *netlist* spécifique de niveau de porte de technologie. Chaque composant est synthétisé et optimisé avec une variété de contraintes jusqu'à ce que la vitesse, la surface occupée, et les caractéristiques désirées de puissance soient obtenues. Ce processus peut exiger un nombre significatif d'itérations quand des contraintes de minimum d'espace ou de très grande vitesse sont appliquées. La règle habituelle est d'écrire un certain code, lancer une simulation fonctionnelle, et obtenir quelques portes synthétisées. Aussi, notre objectif ne sera pas à quelle rapidité nous pouvions obtenir quelques portes mais est à quelle rapidité nous pouvions obtenir toutes les portes correctes.

IX.2.5. Vérification par composants

Toutes nos composantes seront vérifiées individuellement juste après que les évaluations de taille et de vitesse soient produites, c'est-à-dire juste après l'étape de la synthèse. La

vérification composante devrait être un essai approfondi qui inclut des valeurs légales et illégales de chaque état d'entrée, appliqué pour chaque état interne. Ces tests individuels nous permettront de nous concentrer sur les intercommunications composantes plutôt que sur des opérations de vérification de composants.

IX.2.6. Intégration

L'intégration est le processus de prendre les composants examinés et de vérifier qu'ils fonctionnent ensemble. Si les composants ont été exhaustivement examinés, alors ceux-ci intégrés, n'ont pas besoin d'être individuellement analysés. Seulement certaines conditions doivent être vérifiées pour assurer fonctionnement correct d'un composant donné.

Xilinx traduit cette phase par l'implémentation qui se compose des processus de map (traduction de la logique générée en un fichier compréhensible par l'XILINX, c'est-à-dire en composants CLB (voir chapitre précédent), de placement et de routage, pour aboutir à la génération d'un bitstream pour une programmation ultérieure du composant FPGA.

A notre niveau, cette étape nous la ferons au niveau de la simulation en gardant les mêmes sorties pour un composant pour les injecter comme entrées dans celui qui le suit automatiquement.

IX.3. Le code de SOBI en Vhdl

Les résultats que nous vous proposons découlent directement de l'implémentation des algorithmes introduits au niveau du chapitre précédent.

Aussi avant de vous présenter les programmes en tant que tels, nous vous suggérons une petite randonnée pour quelques règles de bases dans la programmation VHDL. Ceci s'impose pour familiariser le lecteur.

IX.3.1. Composition générale d'une description VHDL

Une manière générale de décrire en vhdl n'existe pas. Cependant, des règles générales ont été établies .

Sans pour autant nous étendre, nous dirons que l'organisation d'une conception VHDL se résume suivant la figure (9-2) de la page suivante.

En d'autres termes, une description VHDL admet ces constructions de base suivantes :

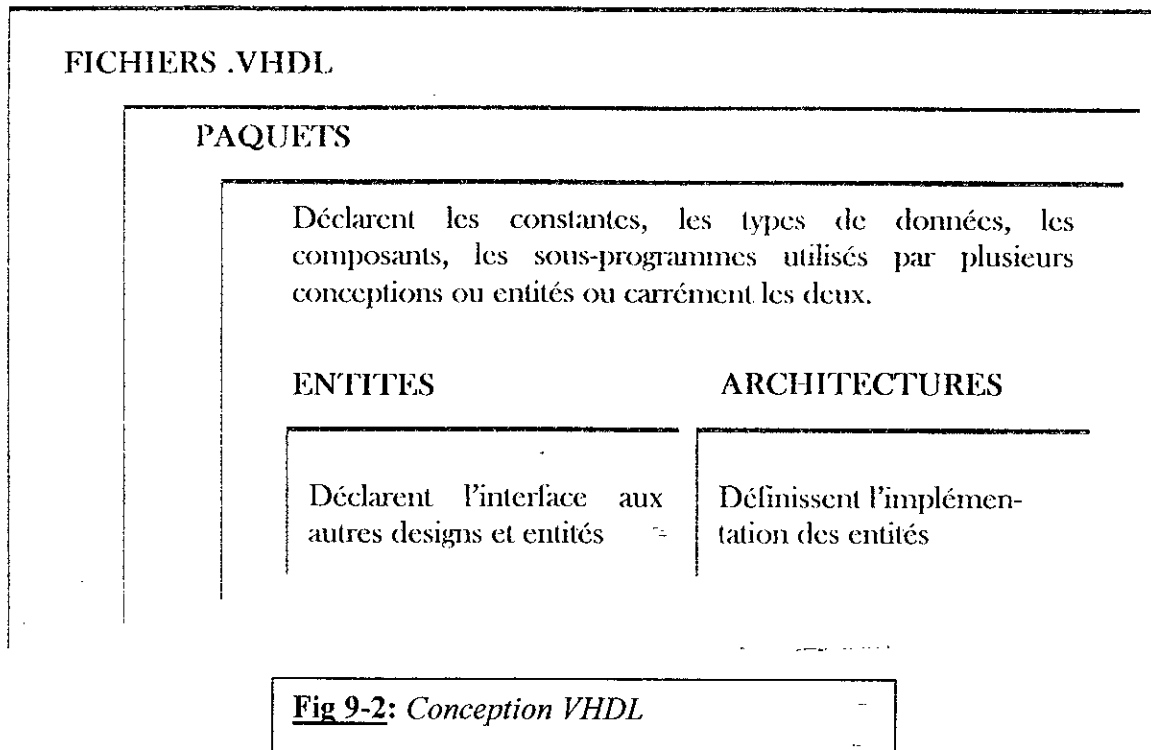
• *L'entité* désigne l'interface d'un niveau pour les autres éléments du projet de conception.

L'entité donne les informations concernant les signaux d'entrées et de sorties de la structure ainsi que leurs noms et leurs types. Elle pointe vers une réalisation complète, un composant hardware, ou un objet VHDL prédéfini. En ce sens alors, une conception VHDL est une concaténation de plusieurs entités.

• *L'architecture* n'est qu'une implémentation de l'entité, traduisant sa fonction. Elle décrit le comportement de l'entité.

Les architectures sont structurées en deux parties essentielles : une première déclarative et une seconde constituée d'une collection de déclarations concurrentes. A ce sujet, nous noterons

cette principale distinction du VHDL relativement aux autres langages évolués tels le C, ou le C++, ou encore le MATLAB, etc. Toutes les instructions s'exécutent simultanément.



- ☞ *Les Configurations* constituent les raccordements des entités aux architectures tandis que
- ☞ *Les Processus* ne sont autres que des collections d'instructions séquentiellement exécutées
- ☞ *Les sous-programmes* sont des algorithmes qui peuvent être appelés par plus d'une architecture.
- ☞ *Les Paquets* désignent des ensembles de déclarations employées par une ou plusieurs conceptions.

A titre d'illustration, nous vous présentons ce petit additionneur deux bits, exemple assez simple, mais spécifique.

Exemple: l'adder

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity additionneur is
  port (
    CLOCK: in STD_LOGIC;
    RESET: in STD_LOGIC;
    ENABLE: in STD_LOGIC;
    INBUSA: in STD_LOGIC_VECTOR (1 downto 0);
    INBUSB: in STD_LOGIC_VECTOR (1 downto 0);
    OUTBUS: out STD_LOGIC_VECTOR (1 downto 0)
  );

```

```

end additonneur;

architecture arch_add of additonneur is

    begin

        process(RESET, CLOCK)
            begin
                if (RESET = '1') then
                    OUTBUS <= "00";
                elsif (CLOCK'event and CLOCK='1') then
                    if ENABLE = '1' then
                        OUTBUS <= INBUSA + INBUSB;
                    end if;
                end if;
            end process;

        end arch_add;
    end

```

IX.3.2. Les outils de développement

Nous utilisons le logiciel *XILINX FOUNDATION F1.5, Student Version* comme outil de validation de notre travail, et appelant pour :

la description

✓ *HDL editor version 3.07.0.39*

✓ *le logiblox (notamment dans l'utilisation des ROM et RAM, des registres de décalages ,....)*

✓ *le logicore (essentiellement dans le calcul de fonctions mathématiques délicates à programmer telle la racine carrée)*

la simulation

✓ *Le Logic Simulator*

la synthèse

✓ *le Synopsys FPGA Macro Compiler version 3.1.1.0w.*

l'implémentation

✓ *le Flow Engine version M 1.5.19*

IX.4. Présentation des résultats

Les programmes relatifs à la conception sont exposés dans la partie annexe. Dans cette partie, il sera question de parler de tous ce qu'on a eu au niveau de la synthèse, de la simulation et de l'implémentation.

IX.4.1. La synthèse

La véritable décision qui pourrait mener à bien ou non l'implémentation se prend à ce niveau. En réalité, la bibliothèque des ressources que l'on aura à appeler à ce stade du développement demande une connaissance assez aiguisée non seulement du niveau de complexité du travail i.e de la description VHDL, autrement dit encore de la forme et du style d'écriture des programmes, que de la connaissances de toutes les capacités disponibles pour chaque bibliothèque afin d'éviter des messages de dépassement de capacités.

Le style de rédaction des programmes joue un rôle non moins négligeable dans ce schéma. Par exemple, l'utilisation de parenthèses qui tend à participer au contrôle de la structure logique : Le chemin le plus long est entre 3 additionneurs dans $Som = A+B+C+D$ alors qu'il n'est qu'à travers 2 additionneurs dans $Som = (A+B) + (C+D)$. Un autre concerne l'emploi de la condition if qui est très utile dans l'encodage des machines à états mais introduit un délai supplémentaire. Et si vraiment, l'application n'a pas besoin de priorité, mieux vaut pas l'utiliser. Dans cette exemple :

```
IF cond_a THEN do_a
ELSIF cond_b THEN do_b
ELSIF cond_c THEN do_c
ELSIF cond_d THEN do_d
ELSE do_e
END IF;
```

pour que la condition do_e occure, il faudra attendre que toutes les autres se réalisent, ce qui correspondra en terme de synthèse à une longue chaîne de retour. Ainsi par exemple, nous pourrions utiliser l'instruction case.

En un mot, notre souci premier était d'incorporer autant que possible toutes ces considérations dans notre design pour obtenir à la fin un optimum de performances.

Eu égard de tout ce qui précède, la décision du choix du composant FPGA n'a pas été facile ; Plusieurs révisions et versions de la plupart des implémentations ont été ainsi effectuées avant d'adopter un composant qui puisse nous offrir de bons résultats et surtout nous implémenter nos programmes.

Aussi, avec l'appui de la référence [DAT 98] et après quelques implémentations avec les autres devices, nous avons pris la décision de travailler avec la famille XC4000E. La famille sollicitée est la **4013XL PQ240**; Car ni le 4003, ni le 4005, ni le 4008, ne nous permettaient une implémentation sans erreurs; Et donc dans la majeure partie de notre design, c'est avec ces options : XC4010E-3-PQ208 .

Précisons que cette écriture signifie simplement :

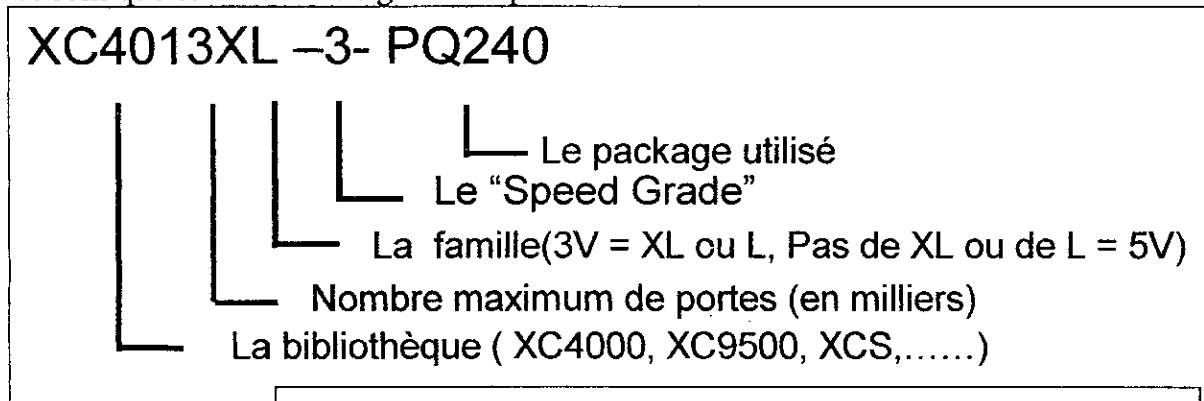


Fig 9-3 : *Convention D'XILINX*

A noter que le « speed grade » désigne une unité de mesure de retard interne; les petits nombres signifient des parties plus rapides pour toutes les familles sauf cependant pour Spartan ou c'est au contraire l'inverse qui se produit.

IX.4.2. La simulation fonctionnelle par composants

L'ensemble des résultats enregistrés dans cette partie est exposé au niveau de l'annexe C ; tous les composants de notre projet sont examinés individuellement pour nous assurer de leur bonne fonctionnalité avant tout traitement d'ensemble.

Ici, nous avons choisi d'en présenter plus en détail un : l'estimée de la matrice d'intercovariance $R(0)$ qui est le point de départ de tout le reste du travail.

Ainsi, nous allons interpréter les résultats du graphe de simulation obtenu à partir du Logic Simulator qui se présentent comme suit :

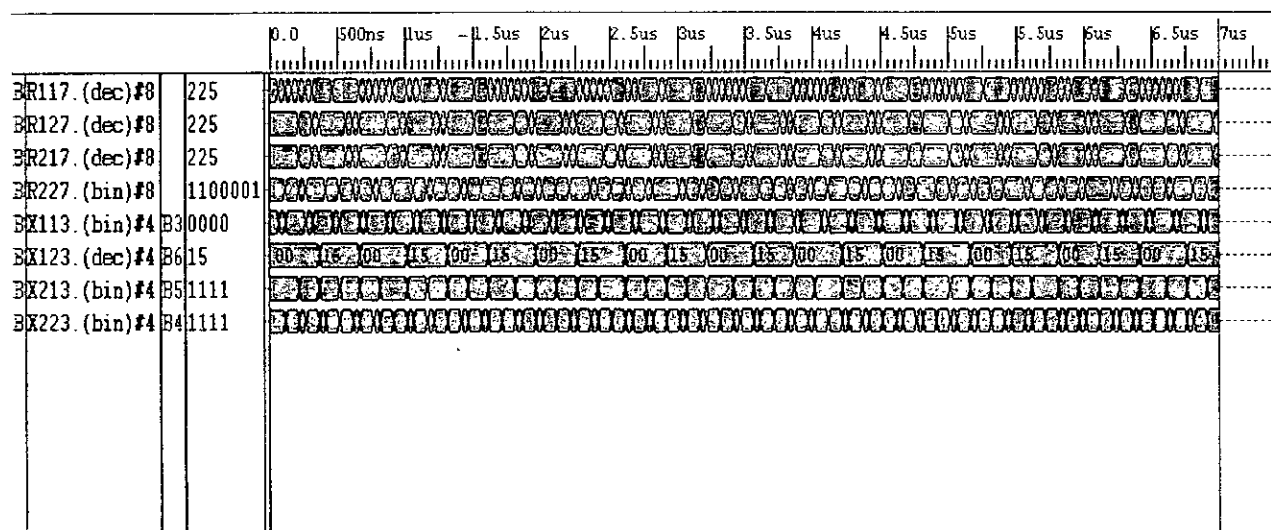


Fig 9-4 : *Waveform relative à l'estimation de $R(0)$*

Analyse des résultats

La partie que nous avons décidée d'introduire ne découle pas d'un simple hasard. En effet, notre choix est plutôt justifié par l'importance de cette matrice dans l'algorithme et par conséquent dans nos implémentations. S'assurer d'abord de sa correcte programmation est une condition sine qua non pour l'obtention de résultats concordants.

Dans le cas où nous avons considéré un nombre de capteurs égal à deux et qui observent un mélange instantané de sources dont le nombre sera supposé égal à deux pour éviter une quelconque modélisation du bruit, et si nous appelons par X11, X12, X21, et X22 les différentes entrées échantillonnées, nous aurons une matrices d'intercovariance des

observations à quatre éléments qui sont R11, R12, R21, R22 régis par les relations exposées dans le code **ro.vhd** dans les annexes.

A titre illustratif, nous ferons la vérification pour le premier élément de la matrice.

Le codage de nos entrées sont faites sur 4 bits et ici, nous aurons des sorties sur 8 bits alors puisqu'en fait, le compilateur n'aura qu'à exécuter cette opération :

$R11 \leftarrow (X11 * X11) + (X21 * X21)$, deux produits plus une sommation.

Et si nous analysons l'instant de simulation qui est présentée ci-dessous, X21 est à 15 et X11 est nul ; ce qui donnera $R11 = 0 * 0 + 15 * 15$ qui est bien égal à 225. Ce qui vérifie bien la correcte implémentation du programme.

Notons tout simplement que nous avons délibérément opté de présenter ces valeurs et en binaires et en décimal pour permettre au lecteur de vérifier avec nous nos affirmations.

Il en est de même pour tous les autres modules créés dans notre projet.

IX.4.3. L'implémentation

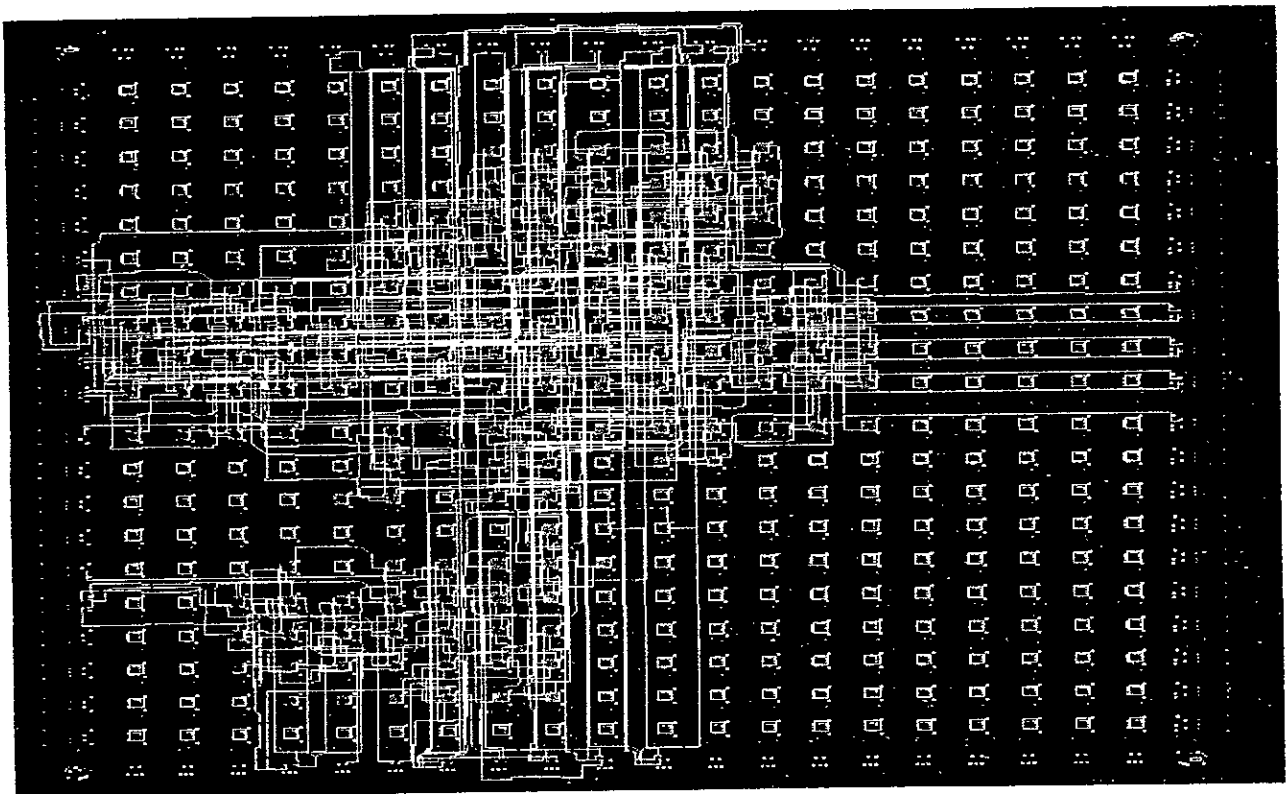


Fig 9-4 : *Circuit correspondant à l'estimée de R(0)*

Le circuit de la feuille précédente est le résultat tacite d'une « traduction » de la logique générée lors de simulation en une logique compréhensible par XILINX, suivi d'un « mapping » où les CLB et les IOB vont entrer en jeu avant d'être « placés » pour un futur routage sur le circuit FPGA. Ensuite et après, les opérations faisant intervenir les contraintes temporelles et la génération du bitstream pourront s'effectuer.

L'unité CLB (Combinatorial Logic Block) peut être comme une mesure de la quantité de ressources logiques dans le composant FPGA.

Aussi, si nous devons commenter ce schéma, il faudra d'abord que nous localisions les entrées et les sorties de par les IOB et les CLB. Ces derniers obéissent à la distribution de la figure (9-5). Les IOB se localisent tout autour constituant l'interface et les CLB tout à fait au centre

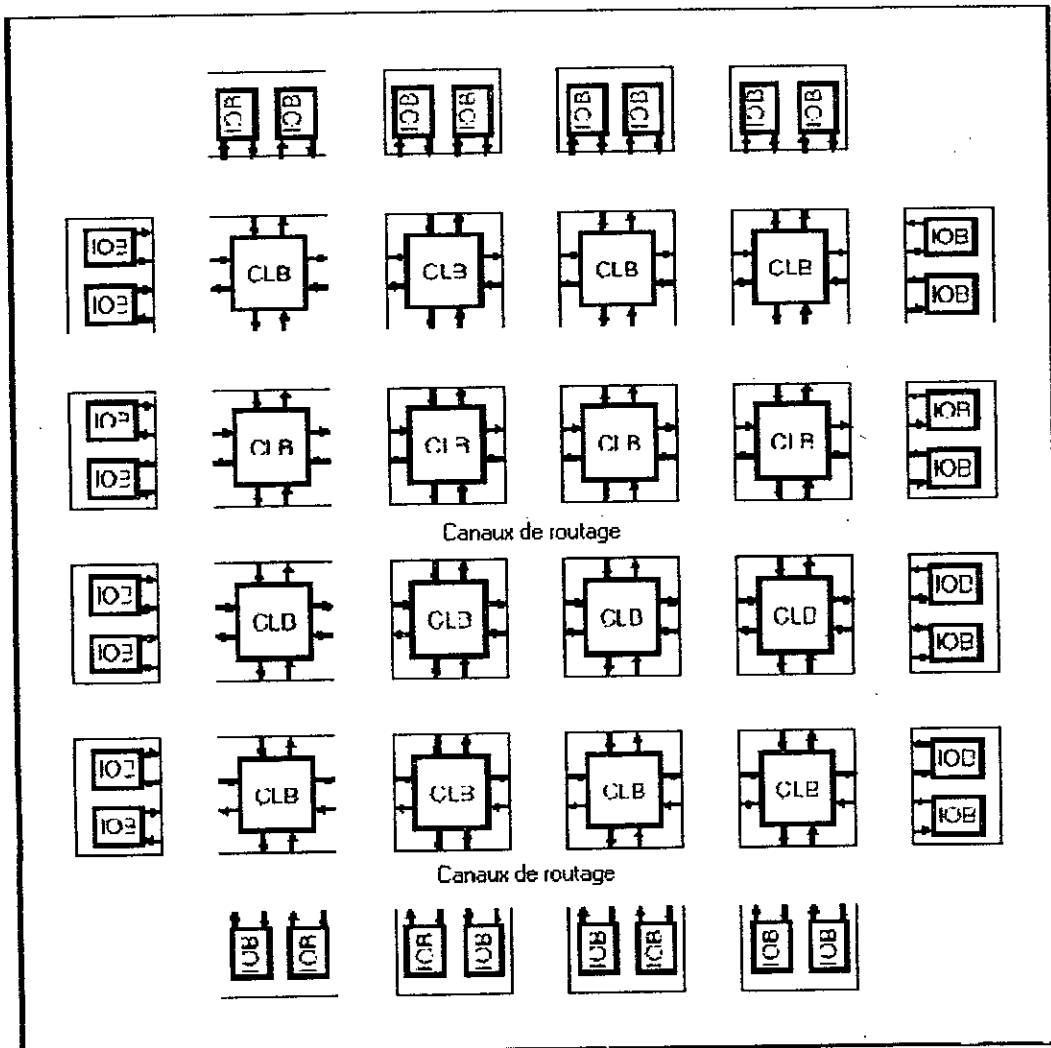


Fig 9-5 : Distribution des CLB et des IOB

Et en exploitant les informations fournies par les « reports » générés lors de l'exécution, nous pouvons résumer les résultats.

Ainsi pour l'estimation de R(0), l'implémentation sur FPGA a utilisé :

- 127 CLB soit un pourcentage de 31% des ressources possibles
- 48 IOB soit un pourcentage de 30% des ressources disponibles

- soit un total de 2196 portes
- avec 6960 chemins, 320 nets, et 468 connections
- et avec un délai maximal de transmission égal à 6,38ns

Déjà, si on se proposait à faire une autocritique concernant le fait de ne pas avoir utilisé même pas la moitié des ressources disponibles, aussi, ne faudrait peut-être travailler avec le 4008 ou bien même avec le 4005 ?

Seulement dans un but d'uniformisation, nous avons préféré garder un même composant FPGA. En effet, rien ne nous empêchait de prendre ces précédents composants, mais ils n'allaient résoudre le problème que partiellement puisque permettant l'implémentation des premiers mais s'avéreraient insuffisants pour les derniers car ne perdons pas de vue que nous sommes en train de manipuler des données à plusieurs dimensions qui ont été modélisées sous forme matricielle et que nous avons finalement codées sous forme binaire pour la réalisation. Aussi, les premières constatations qui en découlent laissent savoir qu'au fur et à mesure qu'on avance dans le design, l'ordre et la taille de nos données allaient augmenter, car SOBI nécessite de capacités de calcul numérique assez importantes qui vont des multiplications matricielles aux calculs à des puissances parfois très élevées, ce qui implique conséquemment des besoins de plus en plus important en ressources pour l'implémentation, ce que le 4005 ou le 4008 ne pouvaient assurer. Même avec des observations codées sur 4 bits, le 4010 n'avait pu achever toute l'implémentation.

IX.5. Les performances

Dans cette section, nous nous proposons d'analyser et de façon plus détaillée nos résultats. Les bonnes performances théoriques sont souvent le résultat d'une étude comparative de la réalisation proposée d'avec une autre jugée comme référence présentant les meilleures approches de l'application conçue ; Or, dans notre cas, une implémentation antérieure de SOBI, ou bien même à défaut, une qui l'approcherait ne serait-ce que pour certaines spécificités communes, sont inexistantes aussi bien dans la littérature que dans la net. Aussi, dans cette étude, nous avons proposé une autre approche d'évaluation. Nous avons implémenté plusieurs versions de l'algorithme en gardant tous les paramètres de synthèse sauf un : la bibliothèque des ressources.

Ce choix n'est pas hasardeux puisque c'est à travers cette décision de la bibliothèque de synthèse que quasiment tous les autres paramètres de performances vont dépendre, aussi bien de la réalisabilité de l'implémentation d'abord que des délais de transmission des signaux, ou de la rapidité de réponse du circuit final, ou même du coût pour la synthèse matérielle.

La XC4010PQ240 ne pouvant achever une implémentation complète, nous nous retournés ve la même famille XC4013 où nous avons réalisé subséquemment 2 versions en gardant un « speed grade » à 3 et tous les autres paramètres constants. Les résultats obtenus sont résumés dans le tableau ci-après.

⊕ Pour la première version

<i>Imp1: XC4013-3-PQ240</i>													
<i>C_{LB}</i>	<i>%</i>	<i>I_{OB}</i>	<i>%</i>	<i>TPL</i>	<i>MD</i>	<i>MDN</i>	<i>MDP</i>	<i>P</i>	<i>N</i>	<i>C</i>	<i>ACDN</i>	<i>ACD</i>	
127	22	48	25	2196	49.831	8.179	8.179	6960	334	501	6.335	2.947	<i>Estro</i>
80	13	64	33	1387	52.333	7.741	7.741	18836	228	368	6.680	3.037	<i>Poly</i>
164	28	64	33	2196	50.001	6.120	7.010	7512	214	752	5.783	3.001	<i>Vecto</i>
165	28	64	33	2652	48.820	7.189	7.189	7408	426	672	6.466	3.042	<i>Matbl</i>
73	12	64	33	1132	36.541	4.963	4.963	1416	206	308	4.368	2.305	<i>Interc</i>
10	1	48	25	194	21.937	4.309	4.309	144	58	60	3.543	2.093	<i>Diago</i>
143	24	56	29	2438	50.359	7.628	7.628	22664	380	612	6.684	3.031	<i>Mel</i>
557	96	112	58	10206	78.217	11.721	11.721	6897392	1456	2374	10.412	3.755	<i>sour</i>

⊕ Pour la deuxième version

<i>Imp2 : XC4013-3-HT144</i>													
<i>C_{LB}</i>	<i>%</i>	<i>I_{OB}</i>	<i>%</i>	<i>TPL</i>	<i>MD</i>	<i>MDN</i>	<i>MDP</i>	<i>P</i>	<i>N</i>	<i>C</i>	<i>ACDN</i>	<i>ACD</i>	
127	22	48	42	2196	59.735	13.703	13.703	6960	337	501	10.456	4.107	<i>Estro</i>
80	13	64	56	1387	57.237	12.492	12.492	18836	228	368	12.492	4.123	<i>Poly</i>
164	28	64	56	2196	57.001	11.225	11.225	7690	501	647	12.696	5.122	<i>Vecto</i>
165	28	64	56	2652	56.004	12.591	12.591	7408	426	672	10.873	4.276	<i>Matbl</i>
73	12	64	56	1132	39.113	7.627	7.627	1416	208	308	5.868	2.804	<i>Interc</i>
10	1	48	42	194	25.216	5.296	5.296	144	58	60	4.452	2.726	<i>Diago</i>
143	24	56	49	2438	57.152	12.995	12.995	22664	379	612	11.349	4.226	<i>Mel</i>
557	96	112	99	10206	88.205	22.229	22.229	6897392	1462	2374	18.450	5.252	<i>sour</i>

Légende :

TPL : Total portes logiques équivalentes

MD : Retard maximum

P : Paths

C :Connections

N : Nets

MDN :retard max pour un net

MDP :Retard max pour un pin

ACDN : retard moyen de connexion pour les 10 plus mauvais nets

ACD :retard moyen de connexion total

La seule conclusion qui puisse découler de cette étude comparative est qu'on a obtenu des retards de transmission plus conséquents dans l'implémentation 2.

Aussi, garderons-nous la première pour un circuit plus rapide.

Cependant, nous nous devons de signaler que l'implémentation réalisée dans tout ce travail n'est un exemple particulier de SOBI. Puisque d'abord, cet algorithme traite de signaux de nature complexe, alors que nous avons travaillé avec des données réelles; Ceci ne change en rien l'exactitude de nos résultats, ni ne les paralyse encore moins, en ce sens que la même procédure peut être conduite pour la partie complexe à quelques changements près.

Un autre aspect est que nos données ont été codées avec un nombre de bits assez petit, indispensable compromis pour pouvoir accéder à l'implémentation (le maximum de CLB à notre disposition est de 576). Ce qui a pour inconvénient majeur de nous limiter dans l'ordre de grandeurs de nos valeurs à introduire.

Les performances obtenues sont assez médiocres si on se propose une comparaison de SOBI implémenté avec un circuit carrément digital manipulant des données d'entrée binaires au départ et qui n'a nullement besoin d'être numérisées.

IX.6. Conclusion

Les résultats obtenus constituent à dire vrai une première dans l'étude d'algorithmes manipulant des données réelles. Aussi, bien que leur véracité aient été prouvées, avec des performances acceptables, leur validation ne pourra se faire qu'après implémentation réelle sur FPGA tout en remarquant qu'avec des bibliothèques plus riches et ressources plus adaptées, d'excellentes implémentations pourraient se faire sous des programmes moins longs et plus élégants.

Conclusion Générale

Tout ce qui a été dit au cours de ce mémoire mérite largement d'être résumé. En effet, il a été développé différentes notions et concepts aussi bien théoriques que pratiques faisant référence à une pluralité de domaines, chacun son importance particulière, la plupart relevant de l'électronique moderne inhérente au traitement de l'information.

Force est de reconnaître, tout d'abord, le caractère actuel de la problématique étudiée, tant son aspect bibliographique que ses prolongements software et hardware. L'algorithme étudié, bien que nécessitant une grande capacité mémoire pour l'exécution des calculs numériques, n'en est pas moins efficace et son importance et sa puissance dans le champ de la séparation aveugle traitant de mélanges linéaires et instantanés n'est plus à démontrer.

Aussi, suivant une méthodologie adoptée, avons-nous exploré ce domaine plus en détails pour en extraire toute la richesse mais également en découvrir toute sa complexité. L'élaboration d'une stratégie de développement axée principalement sur une préalable étude bibliographique s'identifiant par ces trois mots clés : SOBI-VHDL-Implémentation, a été concrétisée par une partie hardware consacrée la description, la simulation, et l'implémentation logicielle. Cette triangularisation du problème et l'adoption d'un cycle de conception nous ont facilité le déroulement des travaux et réduit considérablement la durée de la réalisation.

Et grâce à la flexibilité et à la puissance de nos outils de synthèse, la décomposition de l'algorithme en blocs fonctionnels nous a permis d'avoir une méthode de conception ordonnée et d'avancer régulièrement vers la finalisation du travail, de déceler les problèmes et de les résoudre tout au début de la phase de conception et ainsi qu'une validation progressive au fur et à mesure de l'avancement des travaux.

Il est évident que le *design* de systèmes électroniques aussi complexes implique l'acquisition complète et non fragmentée de tout le bagage pédagogique nécessaire, mais nous pouvons y inclure au sortir de cette conception qu'il relève également de tout un art à part entière.

Le projet SOBI en version implémentée a abouti. Le travail que nous avons réalisé marquera sans doute une des toutes premières en son genre, ce qui s'est justifiée par une diversité et une complexité qui se sont manifestées surtout et finalement par un lot de connaissances nouvellement acquises non négligeables.

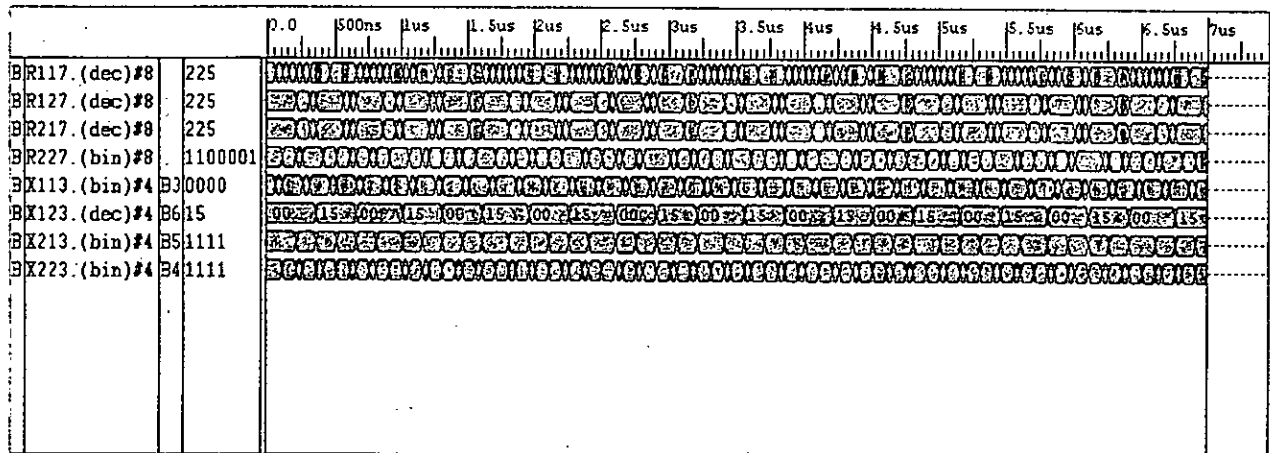
Des perspectives aussi intéressantes que potentielles peuvent être envisageables pour l'avenir. La méthodologie étant établie, il est maintenant possible de mettre au point de nouvelles applications et extensions de notre travail en respectant toutefois les exigences humaines, financières et technologiques.

Il y a lieu de mentionner le plaisir et la passion qui nous ont animés en travaillant sur ce thème, qui est sans doute, d'une cuisante actualité pour nos pays en voie de développement à la recherche d'un essor économique qui passe par la mise en place d'infrastructures adéquates plus particulièrement pour un réveil technologique afin de participer pleinement et d'une autre manière au développement de la science et par là asseoir leur développement.

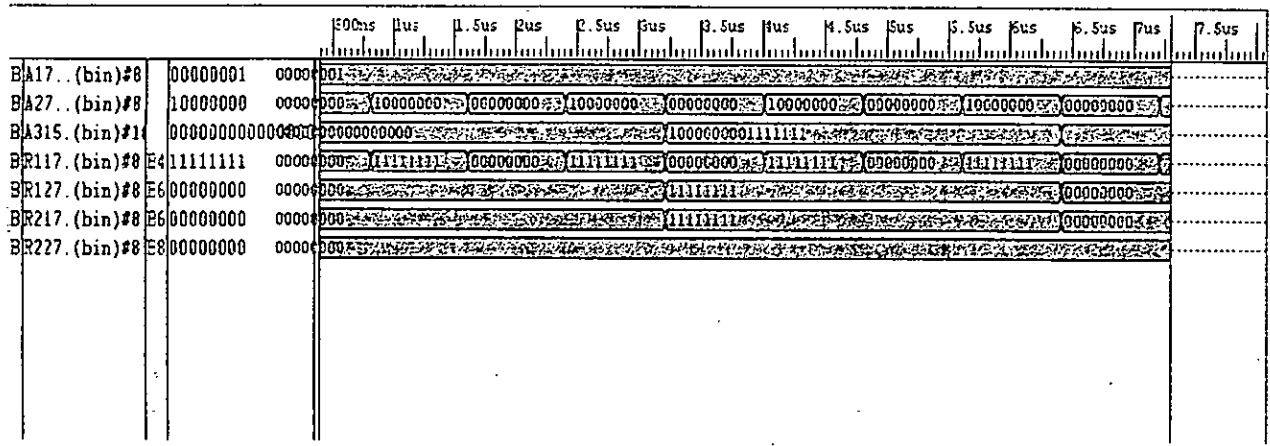
Annexes

Annexe B : Les Simulations

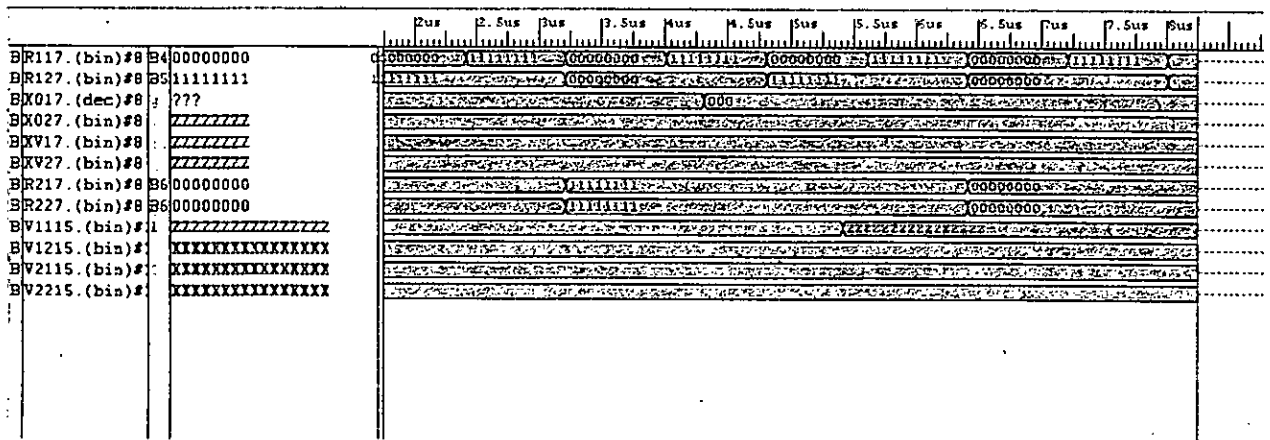
Estimation de R(0)



Le polynôme caractéristique

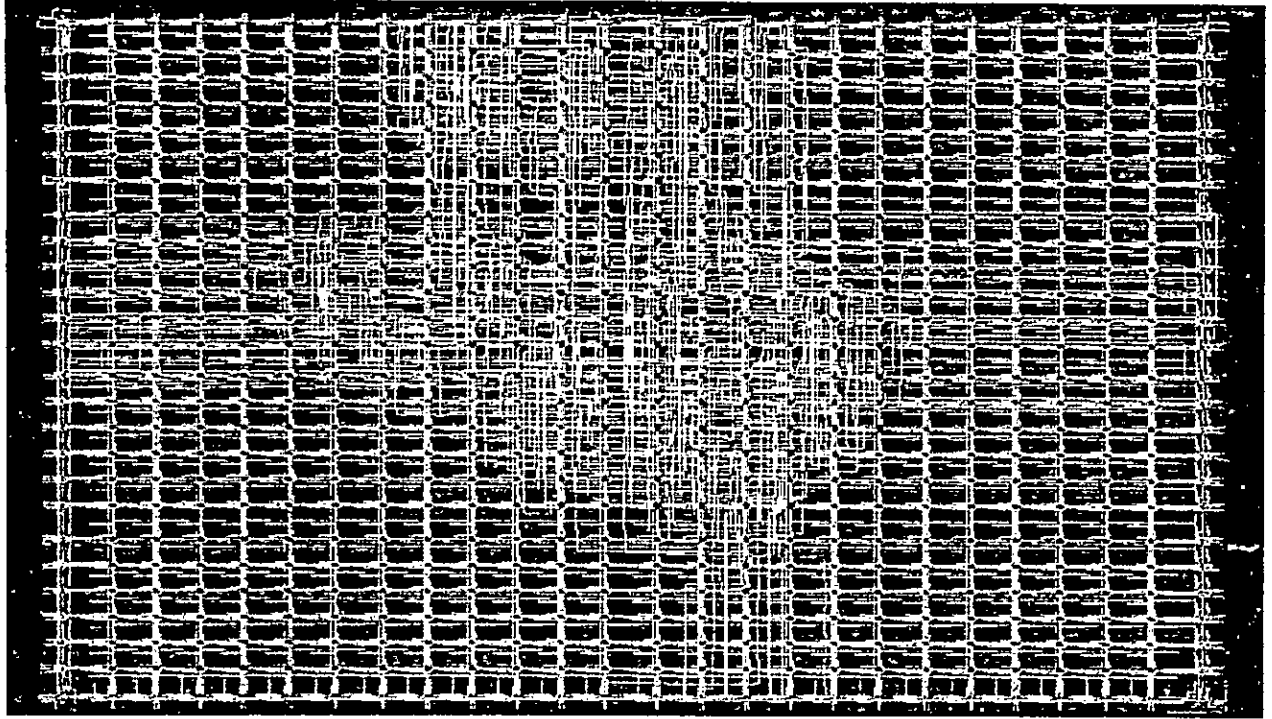


Valeurs et vecteurs propres

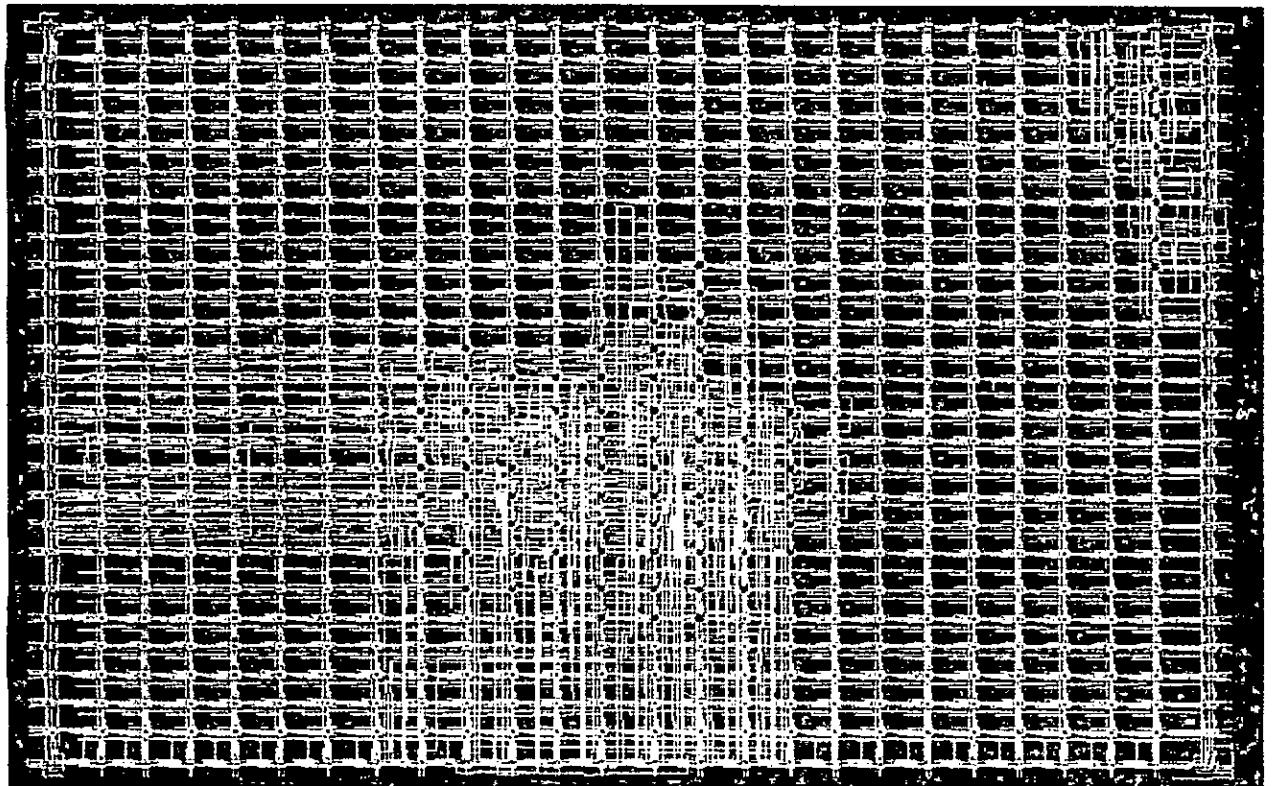


Annexe C : L'implémentation

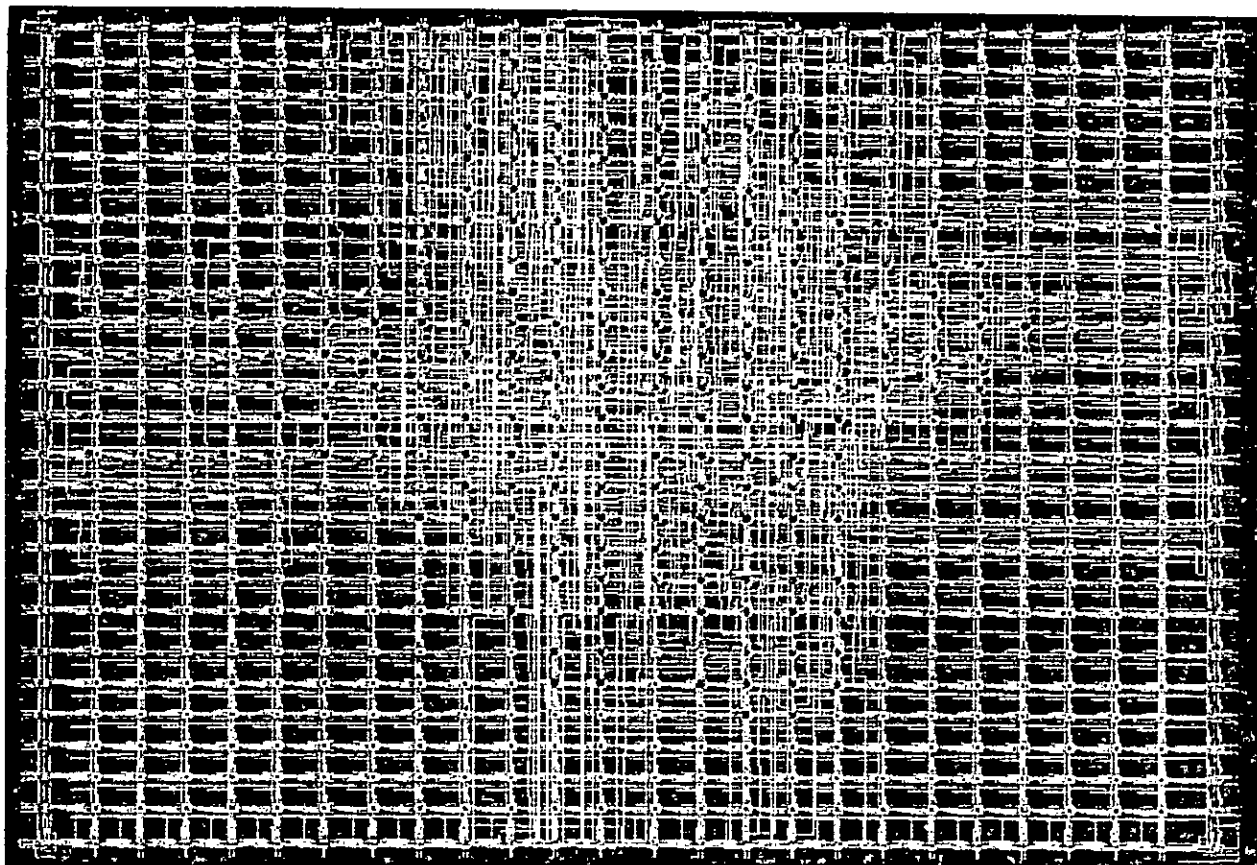
Estimation de $R(0)$



le polynôme caractéristique



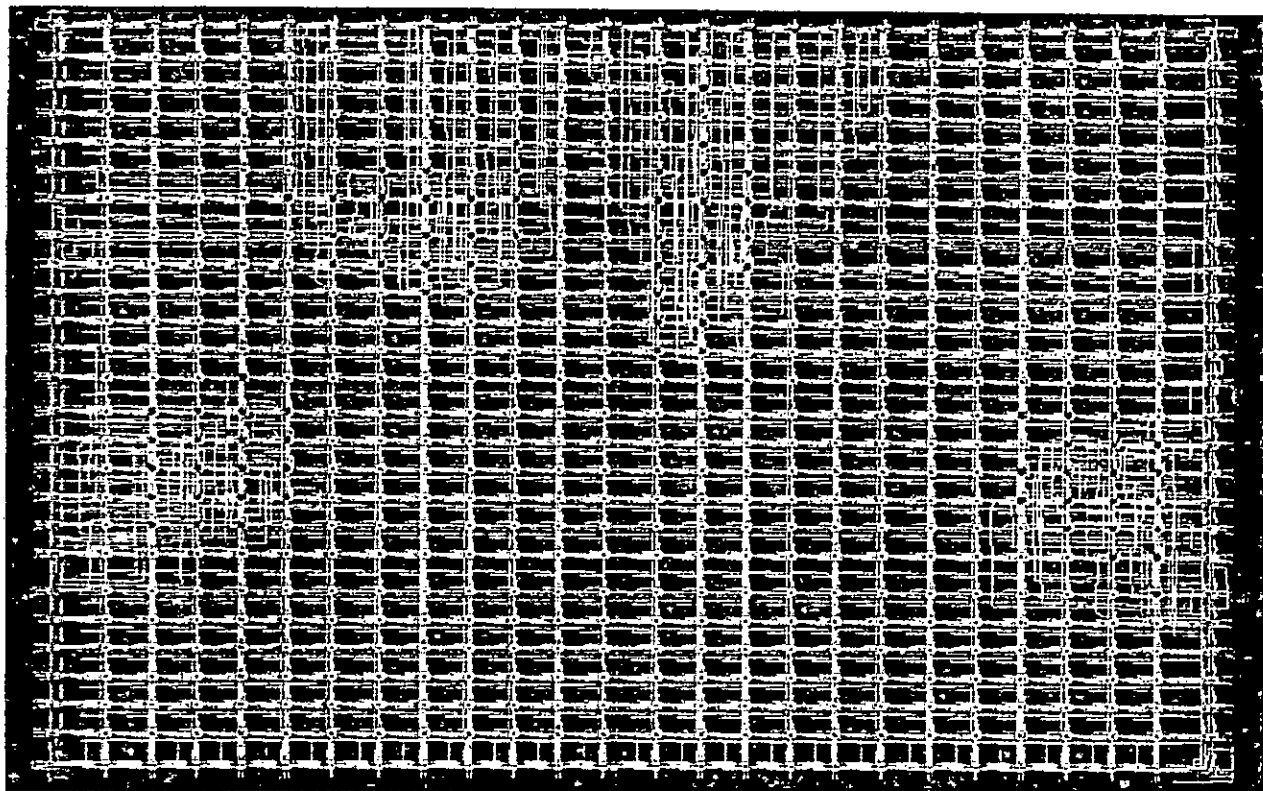
Le blanchiment



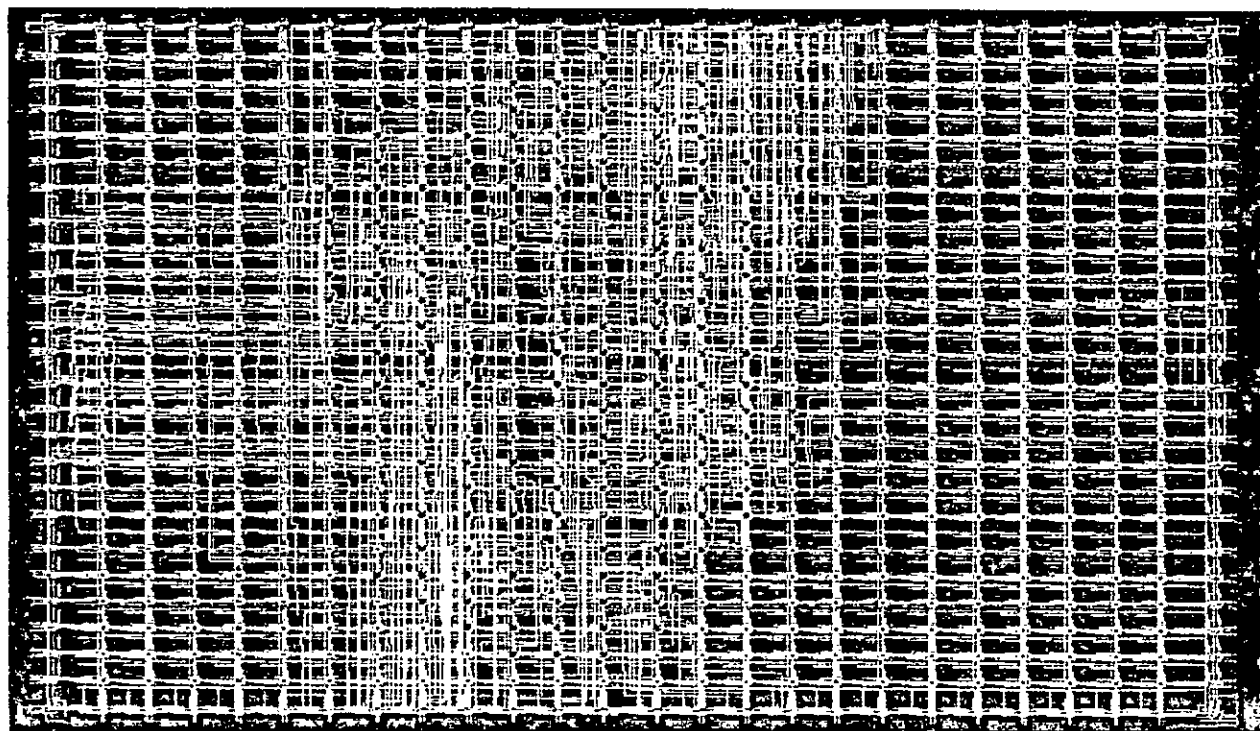
La diagonalisation conjointe



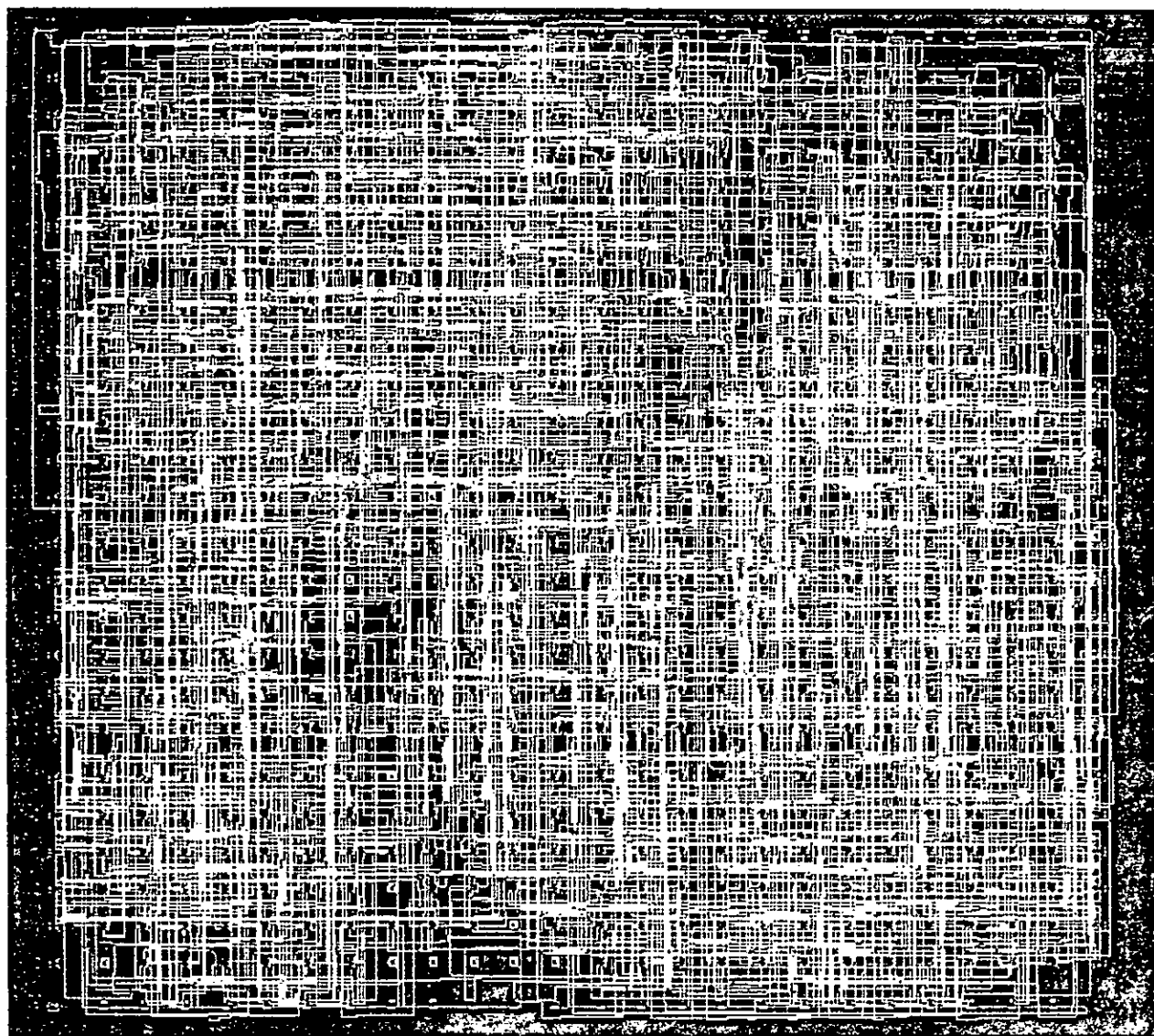
L'intercovariance



Le mélange



Les sources



Bibliographie



[BELOU 95] : A. BELOUHRANI,
"Séparation autodidacte de sources : Algorithmes, Performances, et applications à des signaux expérimentaux", Thèse de doctorat, Ecole Nationale Supérieure des Télécommunications de Paris, 1995.

[BELOU 97] : A. BELOUHRANI, K.ABEID-MAREIM, J-F. CARDOSO, E.MOULINES,
"A blind source separation technique using second order statistics ", IEEE Transactions on Signal Processing, Février 1997.

[ROPE 96] : K.J.ROPE and R.E.BOGNER,
"Blind signal separation, I-linear, instantaneous combinations", Digital Signal Processing, Vol 6, No 1 pp 17-28, 1996.

[APP 00] : A.C. TANG, BARAK A. P, D.B. PHUNG, S.A. CARTER
"Exposé sur la séparation aveugle de sources appliquée à la Magnétoencéphalographie", Part I, Département de Psychologie, d'Informatique, de Neurologie, Université du Nouveau Mexique, Etats-Unis, 28 Juillet 2000

[APP 00] : A.C. TANG, BARAK A. P, D.B. PHUNG, S.A. CARTER
"Exposé sur la séparation aveugle de sources appliquée à la Magnétoencéphalographie", Part II, Département de Psychologie, d'Informatique, de Neurologie, Université du Nouveau Mexique, Etats-Unis, 28 Juillet 2000

[SIB 84] : M.Siboni & J-C.NARDON
"Système linéaires et non linéaires", Hermann, éditeur des sciences et des arts, Pp :III-62,0-49 ;II-119, Décembre 1984.

[Xil 99] : XILINX CORP.
"XC4000E and XC4000X Series Field Programmable Gate Arrays Product Specification." Version 1.5, January 1999.

[DAN 94] : DANIEL D. GASJKI
"A VHDL-based system- design methodology", Department of Information and computer Science, University of California, CA,1994.

[ASH 94] : PETER J. ASHENDEN
"THE VHDL COOK BOOK", Department of Computer Science, University of Adelaide, Australia, première édition, 1990.

[DAT 98] : XILINX CORP.
"The Programmable Logic Data Book", Version 2.0, July 1998.

Nos contacts

www.vhdl.org

www.xilinx.com

—Answers Search - <http://www.xilinx.com/support/searchtd.htm>

—Expert Journals - <http://www.xilinx.com/support/support.htm>

www.dejanews.com

dr_vhdl@vhdl.org

cshelor@acm.org (Charles Shelor, Efficient Networks, Inc.)

lfs@sd.seva.com (Larry Saunders SEVA Technologies Inc.)

sbailey@veribest.com (Stephen A. Bailey VeriBest Inc.)

powell@dqdt.com (Scott R. Powell), dQdt (Dimensions in Quick Design Turnaround)

cesear@dqdt.com (Thomas M. Cesear), dQdt (Dimensions in Quick Design Turnaround)