

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



Ecole Nationale Polytechnique



Département d'Electronique

Laboratoire des Dispositifs de Communication
et de Conversion Photovoltaïque

Thèse de Doctorat en Electronique

Thème

**Algorithmes de Séparation de Sources dédiés
aux Applications Temps réel**

Présentée par **MEZIANE BENTAHAR MEZIANE Abdelfettah**
Magistère en Instrumentation, USTHB

Sous la direction de M. Adel BELOUHRANI Professeur

Présenté et soutenu publiquement le **07 Juin 2017**

Composition du Jury :

Président :	TRABELSI Mohamed	Professeur	ENP
Rapporteur :	BELOUHRANI Adel	Professeur	ENP
Examineurs :	ABED-MERAIM Karim	Professeur	Université Orléans, France
	BELBACHIR Hacene	Professeur	USTHB
	REMRAM Youcef	MCA	USTHB
	ADNANE Mourad	MCA	ENP

ENP 2017

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



Ecole Nationale Polytechnique



Département d'Electronique

Laboratoire des Dispositifs de Communication
et de Conversion Photovoltaïque

Thèse de Doctorat en Electronique

Thème

**Algorithmes de Séparation de Sources dédiés
aux Applications Temps réel**

Présentée par **MEZIANE BENTAHAR MEZIANE Abdelfettah**
Magistère en Instrumentation, USTHB

Sous la direction de M. Adel BELOUHRANI Professeur

Présenté et soutenu publiquement le **07 Juin 2017**

Composition du Jury :

Président :	TRABELSI Mohamed	Professeur	ENP
Rapporteur :	BELOUHRANI Adel	Professeur	ENP
Examineurs :	ABED-MERAIM Karim	Professeur	Université Orléans, France
	BELBACHIR Hacene	Professeur	USTHB
	REMRAM Youcef	MCA	USTHB
	ADNANE Mourad	MCA	ENP

ENP 2017

Remerciements

Cette thèse est le fruit de plusieurs années de recherche effectuées au sein du laboratoire des Dispositifs de Communication et de Conversion Photovoltaïque du Département d'électronique à l'école Nationale Polytechnique d'Alger. Je tiens à remercier toutes les personnes qui ont participé, de près ou de loin, à la réalisation de ce travail de thèse.

Je tiens tout d'abord, à remercier profondément mon directeur de thèse Adel BELOUHRANI pour la qualité de son encadrement, sa disponibilité permanente, sa patience, ses encouragements incessants et ses conseils toujours très pertinents tout au long de cette thèse. Mon aventure dans la recherche, et particulièrement en séparation de sources, a commencé avec vous. Je voudrais ici vous témoigner toute ma reconnaissance pour avoir cru en moi.

Je remercie vivement M. Mohamed TRABELSI pour m'avoir fait l'honneur de présider mon jury de thèse ainsi que les examinateurs M. Karim ABED-MERAIM, M. Hacene BELBACHIR, M. Youcef REMRAM et M. ADNANE Mourad pour avoir bien voulu prendre le temps d'évaluer cette thèse ainsi que pour les remarques pertinentes qu'ils ont soulevées.

J'adresse toute ma gratitude et mes remerciements à M. Aissa-El-Bey Abdeldjalil et M. Chonavel Thierry, Professeurs au département Signal et communication à Télécom Bretagne, France, pour leur accueil chaleureux tout au long de mes séjours au sein de leurs laboratoire. Je me souviendrai toujours des précieuses et passionnantes discussions que nous avons eues ensemble.

Je tiens aussi à remercier l'ensemble des permanents et doctorants de ce département, pour tous leurs conseils et leur bonne humeur. Je remercie spécialement Abdelhakim Youcef pour son aide précieuse et avec qui j'ai eu d'interminables échanges scientifiques très constructifs.

Un immense "Merci" très particulier à mes très chers parents, pour leur amour infini, leur présence, leurs prières et leur grand soutien durant toute ma vie, ainsi qu'à mon cher et très spécial frère et à ma sœur. Je remercie particulièrement mon épouse pour avoir su me rassurer, me comprendre et me soutenir tout au long de cette thèse et mes trois enfants pour avoir supporté mes absences pour l'accomplissement de cette thèse.

Je remercie également l'ensemble de mes anciens enseignants et collègues que j'ai eu l'opportunité de rencontrer à L'USTHB et à l'ENP, Sans oublier ma chère patrie

l'Algérie et ses institutions qui m'ont donné les moyens et la chance de faire des études et d'atteindre la plus haute marche universitaire. Je remercie infiniment mes collègues et amis Slimane, Rabeh, Ghanou, Abdelkrim, Rached, Abdelmadjid, Amine et spécialement à Soufiane et Yasser pour m'avoir donné du souffle quand j'en manquais et pour leur disponibilité à toute épreuve.

Enfin je tiens à remercier infiniment et respectueusement Messieurs : Abdelaziz qui m'a inciter et pousser dans la voie de la recherche et m'a permis d'entamer cette thèse, Karim et Mustapha pour m'avoir encouragé, soutenu et orienté jusqu'à son accomplissement.

ملخص:

الفصل المجرد لمصادر الإرسال يتمثل في استعادة مجموعة من إشارات مصادر الإرسال من خلال مزيج هذه الإشارات دون الحصول على معلومات مسبقة بشأن هيكل الخليط. هذه الأطروحة تقترح عدة خوارزميات حساب تخفيض التكلفة من أجل فصل مصدرين من إشارات مصادر الإرسال التي تستلمها اثنين من أجهزة الاستشعار. الخوارزميات المقترحة تعتبر كلا الحالتين: الضوضاء المضافة البيضاء مكانيا والملونة مكانيا. إصدارات متكيفة من هذه الخوارزميات المطورة مقترحة أيضا. الجزء الأخير من هذه الأطروحة يتناول تنفيذ مشروعين من الخوارزميات في الوقت الفعلي لفصل مصادر الإرسال باستعمال المعالج الرقمي للإشارة TMS320C6713

كلمات مفتاحية: الفصل المجرد لمصادر الإرسال, زرع في الزمن الحقيقي, استخلاص, المعالج الرقمي للإشارة.

Abstract

Blind source separation can be described as the process allowing to separate source signals when being mixed, without any a priori information on the mixture structure. This thesis proposes several low computation cost algorithms for the separation of two single sources based on signals received from two sensors. Both additive noise spatially white and spatially colored are considered in this work. Developed algorithms have been extended to include adaptive processing. The last part of this thesis deals with the implementation of two real-time source separation algorithms on digital signal processing DSP platform (TI TMS320C6713).

Key Words: *Blind Source Separation, Second Order Statistics, Digital signal processor, instantaneous mixtures, Real Time implementation.*

Résumé

La séparation aveugle de sources consiste à récupérer un ensemble de signaux sources à partir de leur mélange sans information a priori sur la structure du mélange. Cette thèse propose plusieurs algorithmes à coût de calcul réduit pour la séparation de deux sources à partir de signaux reçus par deux capteurs. Les algorithmes proposés considèrent les deux cas de bruit additif ; spatialement blanc et spatialement coloré. Des versions adaptatives des algorithmes développés sont aussi proposées. La dernière partie de la thèse traite de l'implémentation en temps réel de deux algorithmes de séparation de sources sur un processeur de traitement du signal DSP, en l'occurrence le TI TMS320C6713.

Mot clés : *Séparation Aveugle de Sources, Statistiques du Second Ordre, processeur de traitement du signal, mélanges instantanés, Implémentation Temps Réel.*

Table des matières

Abréviations

Notations

Liste des figures

Introduction	13
1 Bases de la séparation aveugle de sources	16
1.1 Rétro-perspective historique	16
1.2 Illustration du problème de la séparation aveugle de sources	18
1.2.1 Modélisation du problème "Mélange Linéaire Instantané"	18
1.2.2 Mélanges linéaires convolutifs	19
1.2.3 Mélanges post non linéaires	20
1.2.4 Indéterminations et limitations	20
1.3 Principe de la séparation aveugle de sources	21
1.4 Hypothèses classiques de séparation de sources	21
1.5 Applications potentielles	22
1.6 Conclusion	23
2 Résolution analytique du problème de l'identification aveugle en présence de Bruit spatialement blanc	24
2.1 Solutions analytiques d'identification au deuxième ordre	24
2.1.1 Résolution du système d'équations	25
2.1.2 Matrice de Séparation	28
2.2 Algorithme Analytique pour l'identification aveugle k-ASOBI	29
2.3 Conditions d'identifiabilité	31
2.4 Algorithme Analytique pour l'identification aveugle Optimal k-ASOBI	31
2.5 Simulation et Évaluation des performances	32
2.5.1 Critère de performance	33
2.5.2 Performance de l'algorithme ASOBI	35
2.5.3 Performance de l'algorithme k-ASOBI	35

2.5.4	Performance de l'algorithme Optimal k-ASOBI	35
2.6	Conclusion	40
3	Résolution analytique du problème de l'identification aveugle en présence de Bruit spatialement Coloré	41
3.1	Algorithme Analytique pour l'identification aveugle Robuste ASOBI . .	41
3.2	Algorithme Analytique pour l'identification aveugle Robuste k-ASOBI .	42
3.3	Algorithme Analytique pour l'identification aveugle Robuste Optimal k-ASOBI	43
3.4	Simulation et Évaluation des performances	44
3.4.1	Performance de l'algorithme Robust ASOBI	44
3.4.2	Performance de l'algorithme Robust k-ASOBI	47
3.4.3	Performance de l'algorithme Robust Optimal k-ASOBI	47
3.4.4	Comparaison entre les différentes versions de Robust ASOBI . .	49
3.4.5	Comparaison entre les différentes versions de ASOBI et Robust ASOBI	50
3.5	Conclusion	51
4	Implémentation adaptative	52
4.1	L'algorithme ASOBI Adaptatif	52
4.1.1	Traitement adaptatif efficace	54
4.1.2	Résultats de simulation	55
4.1.3	Évaluation des performances	55
4.1.4	Complexité calculatoire	58
4.2	La version adaptative de l'algorithme K-ASOBI	60
4.3	Conclusion	61
5	Implémentation temps réel sur carte DSK C6713	63
5.1	Notions sur les DSP	63
5.2	Plate-forme de développement	64
5.2.1	L'interface graphique Code Composer Studio	64
5.2.2	Définition du temps réel	66
5.3	Implémentation des algorithmes ASOBI, Robuste-ASOBI et Robuste-k-ASOBI sur carte DSK C6713	67
5.3.1	Évaluation du temps d'exécution et optimisation	67
5.3.2	Validation des trois algorithmes sur simulateur	68
5.3.3	Contrainte d'implémentation en temps réel	72
5.4	Conclusion	78

TABLE DES MATIÈRES

Conclusion Générale	79
Références	81

Abréviations

SAS	Séparation Aveugle de Sources
EEG	Electro-Encéphalo-Gramme
ECG	Electro-Cardio-Gramme
ACI	Analyse en Composantes Indépendante
ACP	Analyse en Composantes Principale
DVP	Décomposition en Valeurs Propres
DVS	Décomposition en Valeurs Singulières
MIMO	Multiple Input Multiple Output
SOBI	Algorithme de séparation au second ordre
ASOBI	Algorithme Analytique de séparation au second ordre
k-ASOBI	ASOBI pour un Retard k arbitraire
Optimal-k-ASOBI	ASOBI pour un Retard k Optimal
R-ASOBI	ASOBI Robust
R-k-RASOBI	ASOBI Robust pour un Retard k arbitraire
Optimal-R-k-ASOBI	ASOBI Robust pour un Retard k Optimal
RSB	Rapport Signal à Bruit
SOS	Statistiques d'Ordre Supérieur
SSO	Statistiques du Second Ordre
DSP	Digital Signal Processing.
FPGA	Field Programmable Gate Array.

Notations

\mathbb{R}	Corps des réels.
\mathbb{C}	Corps des complexes.
h	Les scalaires sont notés par des minuscules.
\mathbf{h}	Les vecteurs sont notés par des minuscules en gras.
\mathbf{h}_i	i -ème élément du vecteur \mathbf{h} .
n	Nombre de sources.
m	Nombre de capteurs.
\mathbf{W}	Matrice de séparation.
\mathbf{H}	Matrice de mélange.
\mathbf{H}_{ij}	(ij) -ème élément de la Matrice \mathbf{H} .
\mathbf{P}	Matrice de permutation.
$\mathbf{S}(t)$	Vecteur des sources de dimension $n \times 1$.
$\widehat{\mathbf{S}}(t)$	Vecteur des sources estimées de dimension $n \times 1$.
$\mathbf{x}(t)$	Vecteur des observations de dimension $m \times 1$.
\mathbf{I}_m	matrice identité de dimension $m \times m$.
\mathbf{D}	matrice bloc-diagonale ou diagonale.
$(.)^T$	Opérateur de transposition.
$(.)^H$	Opérateur de transposition conjuguaison.
$(.)^{-1}$	Opérateur d'inversion.
$E[.]$	Espérance mathématique.
\mathbf{R}_x	Matrice d'autocovariance des observations.
$\mathbf{R}_x(t, \tau)$	Matrice de corrélation d'un vecteur $\mathbf{X}(t)$ de signaux aléatoires.
$\mathbf{R}_m(.)$	Partie réelle.
$\mathbf{I}_m(.)$	Partie imaginaire.
$\det(.)$	Déterminant d'une matrice.
$. $	Valeur absolue.

Liste des figures

1.1	Illustration du problème de la séparation par la soirée cocktail [1].	19
1.2	Structure Générale.	19
2.1	Résultat de la simulation : signaux de parole; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'algorithme ASOBI , Ligne 4 : Signaux Récupérés par l'algorithme k-ASOBI, Ligne 5 : Signaux Récupérés par l'algorithme Optimal k-ASOBI	34
2.2	Résultat de la simulation : signaux EEG et ECG; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'algorithme ASOBI, Ligne 4 : Signaux Récupérés par l'algorithme k-ASOBI , Ligne 5 : Signaux Récupérés par l'algorithme k-ASOBI Optimal	35
2.3	Taux de réjection moyen vs SNR. ASOBI	36
2.4	Taux de réjection moyen vs Taille de l'échantillon. ASOBI	36
2.5	Taux de réjection moyen vs SNR. K-ASOBI	37
2.6	Taux de réjection moyen vs Taille de l'échantillon. K-ASOBI	37
2.7	Taux de réjection moyen vs SNR. Optimal K-ASOBI	38
2.8	Taux de réjection moyen vs Taille de l'échantillon. Optimal K-ASOBI .	39
2.9	Taux de réjection moyen vs SNR. Comparaison des trois algorithmes .	39
2.10	Taux de réjection moyen vs Taille de l'échantillon. comparaison des trois algorithmes	40
3.1	Résultat de la simulation : signaux de parole; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'algorithme Robuste ASOBI , Ligne 4 : Signaux Récupérés par l'algorithme Robuste k-ASOBI, Ligne 5 : Signaux Récupérés par l'algorithme Robuste Optimal k-ASOBI	45
3.2	Résultat de la simulation : signaux EEG et ECG; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'algorithme Robuste ASOBI, Ligne 4 : Signaux Récupérés par l'algorithme Robuste k-ASOBI , Ligne 5 : Signaux Récupérés par l'algorithme Robuste k-ASOBI Optimal	45

3.3	Taux de réjection moyen vs SNR. Robust ASOBI	46
3.4	Taux de réjection moyen vs Taille de l'échantillon. Robust ASOBI . . .	46
3.5	Taux de réjection moyen vs SNR. Robust k-ASOBI	47
3.6	Taux de réjection moyen vs Taille de l'échantillon. Robust k-ASOBI . .	48
3.7	Taux de réjection moyen vs SNR.	48
3.8	Taux de réjection moyen vs Taille de l'échantillon.	49
3.9	Taux de réjection moyen vs SNR.	49
3.10	Taux de réjection moyen vs Taille de l'échantillon.	50
3.11	Taux de réjection moyen vs SNR.	50
3.12	Taux de réjection moyen vs Taille de l'échantillon.	51
4.1	Schéma illustrant l'algorithme ASOBI adaptatif	53
4.2	Résultat de la simulation : signaux de parole ; Ligne 1 : Signaux Ori- ginaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'Algorithme Asobi, Ligne 4 : Signaux Récupérés par l'Algorithme ASO- BIadap1 , Ligne 5 : Signaux Récupérés par l'Algorithme ASOBIadap2 .	55
4.3	Résultat de la simulation : signaux EEG et ECG ; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'Algorithme Asobi, Ligne 4 : Signaux Récupérés par l'Algorithme ASOBIadap1 , Ligne 5 : Signaux Récupérés par l'Algorithme ASOBIadap2	56
4.4	Taux de réjection moyen vs SNR pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$	57
4.5	Taux de réjection moyen vs Taille de l'échantillon pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$	57
4.6	Taux de réjection moyen vs Taille de l'échantillon pour plusieurs lon- gueur du bloc (cas de ASOBIadap2).	58
4.7	Complexité de calcul vs taille de l'échantillon.	59
4.8	Complexité de calcul vs longueur du bloc.	59
4.9	Taux de réjection moyen vs SNR pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$	61
4.10	Taux de réjection moyen vs Taille de l'échantillon pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$	61
5.1	Interface graphique Code Composer Studio.	65
5.2	La carte DSK C6713.	65
5.3	Détail de la carte DSK C6713.	66
5.4	Notion du temps réel. A) Traitement par échantillon. B) Traitement par paquet.	66
5.5	Compilation d'un programme C/C++ avec optimisations.	67

5.6	les signaux mélangés x1 et x2 à implémenter sur carte DSK.	68
5.7	Le diagramme bloc de l'implémentation des algorithmes à implémenter, où se1 et se2 sont les signaux sources estimés.	69
5.8	Diagramme des mélanges avant traitement.	69
5.9	Résultats de traitement après implémentation pour 34000 échantillons avec ASOBI.	70
5.10	Résultats de traitement après implémentation pour 34000 échantillons avec K-ASOBI.	70
5.11	Résultats de traitement après implémentation pour 34000 échantillons avec Robust-ASOBI.	71
5.12	Résultats de traitement après implémentation pour 34000 échantillons avec Robust-k-ASOBI.	71
5.13	limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme ASOBI.	74
5.14	limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme k-ASOBI.	74
5.15	limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme Robust ASOBI.	75
5.16	limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme Robust k ASOBI.	75
5.17	Nombre de cycles d'horloge des quatre algorithmes pour le niveau None.	76
5.18	Nombre de cycles d'horloge des quatre algorithmes pour le niveau Register.	76
5.19	Nombre de cycles d'horloge des quatre algorithmes pour le niveau Local.	77
5.20	Nombre de cycles d'horloge des quatre algorithmes pour le niveau Function.	77
5.21	Nombre de cycles d'horloge des quatre algorithmes pour le niveau File.	78

Introduction

Cette thèse est consacrée au développement et l'implémentation de nouveaux algorithmes de séparation aveugle de sources. Nous traitons dans ce travail le problème suivant : En disposant d'une antenne de m capteurs et observant un mélange linéaire instantané de n sources statistiquement indépendantes entre elles, nous nous proposons de restituer les signaux sources à un facteur près en absence de toute information a priori sur la façon dont ces signaux sources ont été mélangés. La séparation de sources est effectuée en exploitant des informations portant principalement sur les statistiques des signaux lorsque les signaux sources sont temporellement cohérents, il est possible de résoudre le problème en n'exploitant que les statistiques d'ordre deux [2].

Dans cette contribution, nous présenterons plusieurs algorithmes efficaces pour l'identification aveugle d'un système à deux entrées deux sorties. Ainsi, plusieurs solutions sont proposées en exploitant les propriétés de cohérence temporelle des signaux sources [3]. Une fois les coefficients du canal identifiés, la séparation est réalisée grâce à un filtre spatial à "zero forcing" (Annulateur d'interférences). Lorsque l'on connaît d'avance les spécifications d'un problème d'identification aveugle (par exemple le nombre de capteurs et des sources impliqués), nous pouvons concevoir des algorithmes spécifiques de séparation de sources. Il est à remarquer que de telles solutions sont plus appropriées pour des implémentations sur DSP (Digital Signal Processor) ou FPGA (Field Programmable Gate Array) où les itérations sont souvent évitées [4].

Le Filtrage adaptatif a atteint un niveau de maturité lui permettant, aujourd'hui, de prendre une partie importante du traitement statistique du signal. Chaque fois qu'il y a un besoin pour traiter des signaux provenant des opérations dans un environnement avec des statistiques variables dans le temps, le filtrage adaptatif offre une solution attractive et prometteuse [5].

En outre, il est bien connu que lors de l'implémentation sur DSP, les divisions et les racines carrées sont plus complexes à mettre en œuvre que les multiplications et nécessitent plus de ressources en espace mémoire et en temps. Cependant, dans le cas de variation temporelle de l'environnement [6], [5], il est important de développer une solution adaptative au problème de Séparation Aveugle de Sources (SAS) comportant un minimum de divisions et de calcul de racines carrées [7]. Ainsi, nous établirons des

versions adaptatives des solutions d'identification aveugle proposées dans les cas de bruit spatialement blanc et de bruit spatialement coloré.

Ce projet de thèse consiste à proposer des algorithmes à coût réduit de calcul pour la séparation de deux sources à partir de deux capteurs en vue de leur utilisation dans des applications temps réel. Ces algorithmes sont implémentés sur carte DSP selon des architectures judicieusement choisies.

Ce mémoire de thèse comporte cinq chapitres organisés comme suit :

Dans le Chapitre 1, nous présentons une Rétro-perspective historique ainsi qu'une introduction à la problématique de la séparation de mélanges linéaires instantanés de sources. Nous spécifions les hypothèses classiques et les indéterminations inhérentes au problème de la SAS. Nous présentons aussi, dans cette partie, les différents types de mélange. Enfin, les applications potentielles de la séparation aveugle de sources seront présentées.

Dans le Chapitre 2, nous nous sommes consacrés aux différentes approches de Séparation Aveugle de Sources possédant un lien direct ou indirect avec les approches que nous développons dans cette thèse. Ainsi, nous présentons dans un premier temps de nouvelles techniques de séparation de mélanges linéaires instantanés de sources dans le cas où le bruit additif est spatialement blanc. Au cours de ce chapitre, il a été vérifié les conditions d'identifiabilité des algorithmes développés. Ainsi, ces algorithmes simples ont été simulés et les performances observés. Ces algorithmes pourront être facilement implémentables sur les processeurs du traitement du signal.

Dans le Chapitre 3, dans la continuité du travail réalisé au chapitre 2, nous attaquons le problème de l'identification aveugle en présence d'un bruit spatialement coloré. Les algorithmes développés vérifient les conditions d'identifiabilité. Ainsi, les performances montrent l'efficacité de ces algorithmes. De part leur simplicité, ces algorithmes pourront facilement être implémentés sur les dispositifs embarqués et notamment sur les processeurs du traitement du signal DSP.

Dans le Chapitre 4, nous introduisons l'aspect adaptatif de deux algorithmes étudiés aux chapitres précédents. Ces solutions adaptatives efficaces à faible complexité calculatoire et sans aucune division résout le problème de la mobilité et de l'embarquabilité car ces algorithmes sont appropriés pour l'implémentation sur les DSP (Digital Signal Processor) et les FPGA (Field programmable Gate Array) où une division nécessite des ressources importantes.

Dans le Chapitre 5, nous proposons de procéder à une implémentation sur processeur DSP des quatre algorithmes de séparation de sources présentant des performances différentes en prenant en considération les contraintes liées au temps réel. Enfin, nous concluons ce chapitre par quelques exemples concrets d'application

de quatre algorithmes de la Séparation Aveugle de Sources développés dans les chapitres précédents, en particulier, dans les domaines d'application de l'audio (parole et musique).

Enfin, nous terminerons ce manuscrit par une conclusion générale dans laquelle sont résumés les travaux présentés dans cette thèse et sont abordées les perspectives qu'ouvrent ces derniers pour de futurs travaux de recherches prometteurs.

Bases de la séparation aveugle de sources

1.1 Rétro-perspective historique

La séparation aveugle de sources (SAS) consiste à récupérer un ensemble de signaux de sources dans lesquelles, seulement, des mélanges instantanés linéaires sont observés. Cette séparation aveugle devient possible grâce à des diversités existantes entre les signaux sources. Ici, on suppose la diversité spectrale, stipulant que les sources ont des spectres distincts susceptibles de se chevaucher.

Ce problème de séparation de sources a tout d'abord été développé par les travaux de Ch. Jutten, J. Héroult et B. Ans à partir de 1984 [8]. Dans [9], Jutten et Taleb décrivent le problème biologique qui a initié les travaux sur la séparation aveugle de sources. Celui-ci consistait à étudier les réponses musculaires émises à l'issue de différentes sortes d'excitation.

En 1989, J.F.Cardoso [10] propose de considérer les propriétés algébriques des cumulants d'ordre quatre en utilisant l'information d'ordre deux pour rendre unitaire la matrice de mélange, puis de l'identifier par des techniques de diagonalisation. En revanche, il est montré dans [11] que les cumulants d'ordre quatre sont suffisants pour permettre à eux seuls, d'estimer la matrice de mélange.

Puis les travaux de Comon, en 1994, " Independent component analysis, A new concept ?" [12], ont permis de formaliser le lien entre la méthode d'analyse en composantes indépendantes (ICA : Independent component analysis) et le problème de séparation aveugle de sources (BSS) dans le cadre d'un mélange linéaire. En effet, grâce au théorème de Darmois [13], Comon a montré que l'ICA est équivalente à la BSS dans le cadre de mélanges linéaires inversibles et non bruités, à condition qu'il y ait au plus une source gaussienne.

Depuis les années 90, à partir de ces travaux, ont été développées de nombreuses méthodes de résolution du problème de séparation aveugle de sources dans le cadre d'un mélange linéaires menant ainsi, à l'élaboration de nombreux algorithmes. Pour un exposé plus détaillé; nous recommandons de consulter [14] ou [7]. Ceci étant, la

méthode de SOBI (Second Order Blind Identification) développée par A. Belouchrani et al [15], [16], conduit à la séparation en exploitant un critère de diagonalisation conjointe approchée d'un ensemble de matrices de corrélation.

En 1999, Taleb et Jutten [17] ont traité un nouveau type de mélange, les mélanges post non linéaires (PNL). En 2001, Cardoso [10], propose d'envisager "the three easy routes to independent component analysis" pour des mélanges linéaires. A cet effet, partant de la constatation que le problème d'analyse en composantes indépendantes n'est pas équivalent au problème de séparation aveugle de sources dans le cas de signaux gaussiens indépendants et identiquement distribués (i.i.d.), il propose d'envisager trois cas différents, de telle sorte que les deux problèmes d'ICA et de BSS soient équivalents. Ces cas diffèrent par les hypothèses sur le modèle des sources, on peut envisager de ce fait des sources non gaussiennes i.i.d; ou bien des sources gaussiennes non stationnaires, ou enfin, des sources gaussiennes corrélées temporellement. Dans [10], Cardoso décrit les différentes méthodes envisagées, en fonction des hypothèses sur le modèle des sources, afin de résoudre le problème de séparation aveugle de sources, grâce à l'analyse en composantes indépendantes. Enfin, on peut citer aussi, les méthodes dites d'identification aveugle par les outils de l'analyse temps fréquence [18].

La SAS est un problème omniprésent dans de nombreux domaines, tels que la réduction du bruit, traitement radar et sonar, amélioration de la qualité de la parole [19], traitement du signal biomédical [2, 20] et les communications sans fil [7]. Ce problème a été largement traité dans la littérature et de nombreuses solutions efficaces ont été proposées [18]. La SAS des mélanges instantanés a intéressé un grand nombre de chercheurs en raison de ses nombreuses applications [21] et sa souplesse mathématique qui mènent à plusieurs solutions SAS simples [7].

La séparation de sources est effectuée en exploitant des informations portant principalement sur les statistiques des signaux [10]. Lorsque les signaux sources sont temporellement cohérents, il est possible de résoudre le problème en n'exploitant que les statistiques d'ordre deux, mais si les signaux sources sont temporellement blancs ou présentent des densités spectrales normalisées identiques, en absence d'informations a priori sur la distribution des sources, toute solution nécessite l'utilisation des statistiques d'ordre supérieur à deux des signaux reçus. Lorsque les distributions des sources sont connues, une solution du maximum de vraisemblance devient possible. Signalons que dans le cas de signaux sources temporellement blancs et gaussiens, le problème de la séparation aveugle de sources n'admet pas de solution. Toutefois, les méthodes de séparation de sources exploitant les statistiques d'ordre supérieur tolèrent au plus une seule source gaussienne [18].

Le choix des méthodes fondées sur les statistiques d'ordre deux est motivé par la capacité de séparer des sources gaussiennes et la prise en compte de la corrélation temporelle et la réduction de la complexité arithmétique [22]. Pour des spécifications particulières, il est possible de concevoir un algorithme SAS spécifique. Pour le

présent problème, le nombre des capteurs et des sources [23] sont spécifiés. Ensuite, des solutions analytiques (closed form solutions) deviennent possibles. Ceci est, spécialement, vrai pour un canal à deux entrées et deux sorties [24, 25]. En outre, il est bien connu que les divisions et les racines carrées sont plus complexes à implémenter que les multiplications et nécessitent plus de ressources spatiales et temporelles.

Dans le cas des environnements variables dans le temps [6], il est important de fournir une solution adaptative aux problèmes de SAS [26] avec un minimum de calcul des divisions et des racines carrées [8].

1.2 Illustration du problème de la séparation aveugle de sources

Il est courant d'observer en milieu naturel des mélanges de signaux provenant de sources différentes. Le célèbre problème de la "soirée cocktail" (effet cocktail party) évoque le cas d'une soirée où les voix des convives se mélangent allègrement [1]. Pourtant, chacun a déjà constaté l'extraordinaire capacité de l'ouïe humaine à différencier l'une de ces voix en particulier, celle de leur interlocuteur par exemple (figure 1.1). Cette capacité peut en effet être qualifiée d'extraordinaire, lorsque l'on constate que l'ouïe humaine est capable d'effectuer cette discrimination dans des conditions extrêmes, que ce soit en présence de très nombreuses sources, ou encore lorsque le bruit ambiant est bien supérieur à la voix que l'on cherche à discerner. Et bien souvent, la création d'Allah réalise avec une facilité déconcertante cette tâche qui devient très ardue dès que l'on souhaite la réaliser artificiellement. Ce problème entre dans le cadre plus général de la séparation aveugle de sources qui consiste à retrouver un certain nombre de sources à partir des observations d'un mélange de celles-ci.

Présenté ainsi dans son contexte le plus général, le problème est très difficile à résoudre pour une machine. Pourtant, ce formalisme permet de modéliser les difficultés rencontrées dans de nombreuses applications. Le problème de la séparation de sources consiste donc, à concevoir des méthodes capables de retrouver les N sources inconnues observées aux travers de M mélanges inconnus des N sources. Ces M mélanges sont obtenus par un réseau de M capteurs.

1.2.1 Modélisation du problème "Mélange Linéaire Instantané"

Le problème de la "cocktail party" est une illustration de la situation dans laquelle la séparation aveugle de sources permet d'obtenir des résultats souvent très convaincants. Dans une pièce où sont présentes deux personnes qui parlent indépendamment les unes des autres, nous plaçons aussi deux magnétophones. Il est naturel de penser que sur les bandes des deux cassettes, on va pouvoir entendre les voix des deux personnes superposées les unes sur les autres. L'objectif de la séparation aveugle est alors de retrouver les paroles des deux personnes à partir seulement des 02 enregistrements des magnétophones (voir figure 1.1) [1].

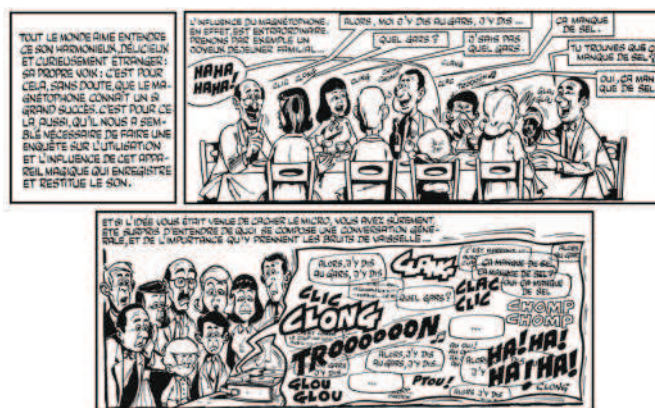


Figure 1.1 — Illustration du problème de la séparation par la soirée cocktail [1].

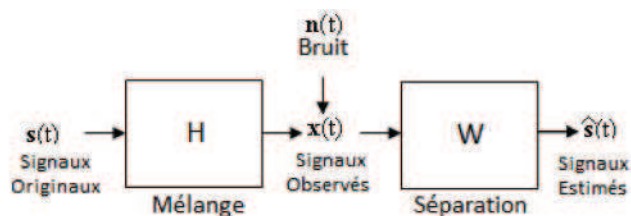


Figure 1.2 — Structure Générale.

Nous considérons un réseau de deux capteurs recevant des signaux à partir de deux sources. La sortie du réseau est notée par $x_i(t), i = 1, \dots, 2$. Celle-ci est corrompue par un bruit blanc additif, noté $\mathbf{n}(t)$ (voir figure 1.2). L'observation est classiquement modélisée par :

$$\mathbf{x}(t) = \mathbf{H} \mathbf{s}(t) + \mathbf{n}(t), \quad (1.1)$$

Où $\mathbf{s}(t)$ est un vecteur 2×1 dont les éléments sont les sources $\mathbf{s}_i(t), i = 1, \dots, 2$. La matrice de mélange \mathbf{H} , dont les éléments sont notés h_{ij} , dépendant uniquement du canal et des capteurs, est donnée par

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}$$

La matrice \mathbf{H} est supposée de rang plein et les signaux sources sont du second ordre stationnaires, temporellement colorés et mutuellement décorrélés [21]. Le but de la séparation de sources est de restaurer les signaux source à partir des signaux de sortie $\mathbf{x}(t)$ sans la connaissance de la matrice de mélange \mathbf{H} . La solution de séparation de sources basée sur des statistiques de second ordre suppose que les signaux source sont décorrélés. En outre, nous supposons que le bruit $\mathbf{n}(t)$ est spatialement blanc et non corrélé avec les signaux source.

1.2.2 Mélanges linéaires convolutifs

Pour les mélanges linéaires convolutifs, les observations sont des mélanges linéaires de signaux issus des sources par filtrage [11]. Le modèle en temps continu est :

$$\mathbf{x}_i(t) = \sum_{j=1}^N \mathbf{h}_{ij} * \mathbf{s}_j(t) + \mathbf{n}_i(t) \quad i = 1, \dots, 2 \quad (1.2)$$

Où \mathbf{h}_{ij} est la réponse impulsionnelle entre la j^{eme} source et le i^{eme} capteur, et l'opérateur $(*)$ correspond au produit de convolution.

1.2.3 Mélanges post non linéaires

Dans de nombreuses applications, les mélanges linéaires instantanés ou convolutifs sont pris par défaut par exemple lors de saturations de capteurs. Taleb et Jutten [17] ont traité les mélanges instantanés post non linéaires (PNL). Le modèle se déroule en deux étapes, la première étape est un mélange linéaire instantané. Dans la seconde étape, chaque observation passe par une fonction non linéaire. Le modèle PNL s'écrit :

$$\mathbf{x}_i(t) = f_i\left(\sum_{j=1}^N \mathbf{h}_{ij} \mathbf{s}_j(t) + \mathbf{n}_i(t)\right) \quad i = 1, \dots, 2 \quad (1.3)$$

Dans cette thèse, on ne considère que le cas de mélange linéaire instantané.

1.2.4 Indéterminations et limitations

On peut observer que la représentation (1.2.1) n'est pas unique. Dans le contexte de la séparation aveugle de sources, l'identification exacte de la matrice du mélange \mathbf{H} est impossible, comme illustré par la relation suivante :

$$\mathbf{x}(t) = \mathbf{H} \mathbf{s}(t) + \mathbf{n}(t) = \sum_{i=1}^2 \frac{\mathbf{h}_i}{\alpha_i} \alpha_i s_i(t) + \mathbf{n}(t) \quad (1.4)$$

où $\alpha_i \in \mathbb{R}^*$ et \mathbf{h}_i dénote la i^{eme} colonne de \mathbf{H} . Par conséquent, l'échange d'un facteur scalaire fixe entre le signal source et la colonne correspondante de \mathbf{H} n'affecte pas les observations. Un avantage peut être tiré de cette indétermination sans aucune perte de généralité, en supposant une variance unité des signaux source, de sorte que la dynamique de la source est représentée par l'amplitude des colonnes correspondantes de \mathbf{H} . On note aussi que la numérisation des sources est arbitraire [18].

Par conséquent, l'identification aveugle de \mathbf{H} se rapporte à la détermination d'une matrice \mathbf{H} à un facteur et une permutation près de ses colonnes. Le point crucial est que ces indéterminations n'entravent pas la séparation de sources. Si la matrice \mathbf{H} est estimée à un facteur et une permutation près de ses colonnes, elle permet encore de déterminer les signaux source à la permutation correspondante et le facteur d'échelle [27]. Dans cette thèse, nous exploitons ces indéterminations pour développer des algorithmes de SAS spécifiques sans divisions.

Dans la littérature, nous trouvons aussi d'autres stratégies pour lever l'indétermination sur les puissances des sources qui consistent, soit à :

- Fixer les éléments diagonaux de la matrice séparante à 1 [28]
- Fixer les éléments diagonaux de la matrice de mélange \mathbf{H} à 1 [8]
- Fixer la norme de chaque vecteur colonne de la matrice séparante à 1 [29]

Nous proposons au chapitre 3, une autre stratégie pour lever cette indétermination sur les puissances des sources.

Finalement, pour résumer, nous appellerons "séparation de sources" l'opération qui, à partir des signaux capteurs, nous permet d'obtenir un ensemble de signaux proportionnels aux signaux émis par chacune des sources. De même la matrice de mélange ne peut être estimée qu'à un facteur et à une permutation près de ses colonnes.

1.3 Principe de la séparation aveugle de sources

Le principe général de la séparation de sources consiste à appliquer des transformations aux observations, pour obtenir des signaux statistiquement indépendants qui seront proportionnels aux signaux sources qu'on veut estimer. Ainsi, le problème de la séparation de sources revient à celui de la recherche de composantes indépendantes, dans un mélange linéaire de ces dernières. Nous serions tentés de penser, comme première solution, à rendre les observations indépendantes à l'ordre deux en exploitant la matrice d'autocovariance de ces dernières, mais on s'est rendu compte rapidement dans [18], que celle-ci est insuffisante pour identifier la matrice de mélange. En effet, la matrice d'autocovariance des observations est donnée par :

$$\mathbf{R}_x = \mathbb{E}[x(t)x(t)^*] = \mathbf{H}\mathbf{H}^H + \mathbf{R}_b \quad (1.5)$$

En supposant que les signaux sources soient de puissance unité : $\mathbf{R}_s = \mathbf{I}_n$

Certaines techniques de séparation de sources commencent d'abord par utiliser la matrice d'autocovariance des observations pour déterminer la matrice \mathbf{H} à une matrice unitaire, ensuite, elles tendent à chercher cette rotation manquante en utilisant d'autres informations. Dans le cas où les signaux sources présentent une cohérence temporelle, nous pourrions utiliser les matrices d'intercovariance des observations pour déterminer la matrice unitaire manquante. Dans le cas où les signaux sources sont non gaussiens, on utilise des cumulants comme outils de test d'indépendance [18] afin d'obtenir des signaux statistiquement indépendants à un ordre supérieur à deux. Si les distributions des sources sont connues, le problème de la séparation de sources devient celui d'une estimation des paramètres précédents au sens du maximum de vraisemblance [16].

1.4 Hypothèses classiques de séparation de sources

La séparation est dite aveugle (non-supervisée ou bien autodidacte) si on est capable de séparer les signaux sans aucune information sur le système de mélange et que les sources sont inobservables. Sans hypothèses supplémentaires la SAS apparaît comme un problème insoluble. C'est pourquoi la plupart des techniques de SAS reposent sur l'hypothèse d'indépendance des sources, laquelle implique leur décorrélation, c'est à

dire que la fonction d'intercorrélation entre deux sources $\mathbf{s}_i(t)$ et $\mathbf{s}_j(t)$ avec $i \neq j$ est nulle pour tout retard τ sachant que $\mathbf{R}_{ij}(t, \tau) = E[s_i(t)s_j^*(t+\tau)] = 0$ ¹. En fonction des méthodes, un certain nombre d'hypothèses autres que l'indépendance s'avèrent parfois nécessaires au bon fonctionnement des méthodes de SAS [30] :

1. Les composantes du vecteur des sources $\mathbf{s}(t)$ sont temporellement colorées, ce qui signifie que la fonction d'intercorrélation entre une source et sa version retardée ou bien entre deux versions retardées de cette source est non-nulle.
2. Le vecteur des sources $\mathbf{s}(t)$ est stationnaire au sens strict et ergodique, cette hypothèse trouve son intérêt lorsque certaines quantités statistiques (telles que les cumulants, les covariances, . . .) doivent être estimées à partir d'une seule réalisation d'un processus aléatoire. Généralement, l'estimateur statistique non-biaisé est consistant dans le contexte stationnaire ergodique. Cependant, du point de vue pratique ou plus précisément pour des signaux de parole, des signaux issus de télécommunications, de machines tournantes ou d'engrenages, les signaux biologiques et physiologiques (ElectroEncéphaloGramme (EEG), Electro-Cardio-Gramme (ECG), Electro-Myo-Gramme (EMG), Magnéto-Encéphalo-Gramme (MEG), température, . . .), . . ., cette hypothèse n'est pas toujours vérifiée.
3. Dans le cas d'un système de mélange sur-déterminé, le nombre de capteurs m est supérieur au nombre de sources n .
4. Dans le cadre des mélanges convolutifs (resp. instantanés), la matrice \mathbf{H} obtenue après reformulation du modèle convolutif est de rang plein, c'est à dire $\text{rang}(\mathbf{H}) = N$, ceci garantit l'existence d'une inverse à gauche de la matrice \mathbf{H} , et donc l'existence du séparateur.
5. Le vecteur $\mathbf{n}(t)$ contenant les bruits est stationnaire. Cette hypothèse ne trouve un intérêt que d'un point de vue pratique (comme pour l'hypothèse 1), en d'autres termes elle n'est pas nécessaire en théorie.
6. Les composantes de $\mathbf{s}(t)$ sont statistiquement indépendantes des composantes de $\mathbf{n}(t)$.
7. Les composantes de $\mathbf{n}(t)$ sont blanches mutuellement indépendantes, de moyenne nulle et de variance σ_n^2 . La gaussianité permet aux méthodes utilisant des cumulants d'ordre strictement supérieur à deux d'être insensibles aux bruits gaussiens.

1.5 Applications potentielles

La séparation de sources présente un grand intérêt pour de nombreuses applications qui se focalisent présentement sur trois domaines : le biomédical [31], l'audio [32] et les télécommunications [33]. Un des avantages majeur de la séparation de sources est de ne pas nécessiter la connaissance d'un signal de référence, comme il est d'usage dans les méthodes classiques de traitement du signal. En outre :

- Pour séparer des signaux biomédicaux (EEG, ECG, . . ., etc.) [34] [35], les capteurs de surface sont suffisants, et évitent ainsi d'utiliser des méthodes invasives pour placer des électrodes sur les sources de référence.

1. On suppose ici que les signaux sont de moyenne nulle.

- Des signaux de brouillage, même ressemblants aux signaux utiles, peuvent être éliminés sans problème particulier [36] [37].
- Le diagnostic des défauts mécaniques dans des machines tournantes utilisant l'analyse vibratoire [38].

1.6 Conclusion

Nous avons vu dans ce chapitre que le problème de la séparation de sources revient souvent à celui de la recherche de composantes indépendantes dans un mélange linéaire. Par ailleurs une identification complète de la matrice de mélange est impossible dans un contexte aveugle. La matrice de mélange ne peut être estimée qu'à un facteur et une permutation près de ses colonnes.

Certaines méthodes de séparation aveugle de sources procèdent d'abord par l'identification aveugle du mélange ensuite restituent les signaux sources par des procédures de filtrage spatial. Dans un contexte aveugle l'hypothèse de base en séparation de source est l'indépendance statistique des signaux sources. Cette hypothèse ne suffit pas à elle seule pour résoudre le problème de la séparation de sources. Il est nécessaire alors de disposer d'une hypothèse supplémentaire.

Dans ce qui suit, nous nous baserons sur les méthodes de séparation de sources exploitant la cohérence temporelle des sources, donc, utilisant uniquement les statistiques d'ordre deux en s'intéressant à trouver des algorithmes simples en vue de leur implémentation sur des circuits hardwares.

Résolution analytique du problème de l'identification aveugle en présence de Bruit spatialement blanc

Dans ce chapitre, nous proposons un nouvel algorithme efficace pour l'identification aveugle d'un système à deux entrées deux sorties dans le cas où le bruit additif est spatialement blanc. Par la suite, plusieurs variantes sont développées.

2.1 Solutions analytiques d'identification au deuxième ordre

Reprenons la version des données échantillonnées du modèle (1.1),

$$\mathbf{x}(n) = \mathbf{H} \mathbf{s}(n) + \mathbf{n}(n), \quad \mathbf{s}(n) = [s_1(n) \ s_2(n)]^T \quad (2.1)$$

Avec les hypothèses formulées dans la section 1.2.1 et $\mathbf{x}(n) = [x_1(n) \ x_2(n)]^T$, $\mathbf{x}_i = [x_i(1), \dots, x_i(T)]^T$ ($i = 1, 2$) les échantillons du vecteur observé sur le capteur i , $\mathbf{s}_i = [s_i(1), \dots, s_i(T)]^T$, $\mathbf{R}_{\mathbf{s}_i \mathbf{s}_j} = \mathbb{E}[\mathbf{s}_i \mathbf{s}_j^T]$.

Nous pouvons alors écrire les observations sous la forme :

$$x_1(n) = h_{11}s_1(n) + h_{12}s_2(n) + n_1(n), \quad (2.2)$$

$$x_2(n) = h_{21}s_1(n) + h_{22}s_2(n) + n_2(n), \quad (2.3)$$

avec

$$\mathbb{E}[\mathbf{n}_i(n) \ \mathbf{n}_j(n - k)] = \mathbb{E}[\mathbf{n}_i(n) \ \mathbf{n}_j(n)]\delta(k) \quad (2.4)$$

Où $\mathbb{E}[\cdot]$ désigne l'opérateur de l'espérance mathématique, sachant que nous sommes dans le cas d'un bruit temporellement blanc. Les statistiques exactes et celles estimées sont sous la forme :

$$\mathbf{r}_{\mathbf{x}_i \mathbf{x}_j}(k) = \mathbb{E}[\mathbf{x}_i(n) \mathbf{x}_j(n-k)] \quad (2.5)$$

$$\hat{r}_{\mathbf{x}_i \mathbf{x}_j}(k) = \frac{1}{N} \sum_{n=k}^{N-1} x_i(n) x_j(n-k) \quad (2.6)$$

Où N : nombre d'échantillons, k : retard temporel. On obtient alors le système d'équations suivant :

$$\mathbf{r}_{\mathbf{x}_1 \mathbf{x}_1}(k) = h_{11}^2 \mathbf{r}_{\mathbf{s}_1 \mathbf{s}_1}(k) + h_{12}^2 \mathbf{r}_{\mathbf{s}_2 \mathbf{s}_2}(k) + \sigma^2 \delta(k), \quad (2.7)$$

$$\mathbf{r}_{\mathbf{x}_2 \mathbf{x}_2}(k) = h_{21}^2 \mathbf{r}_{\mathbf{s}_1 \mathbf{s}_1}(k) + h_{22}^2 \mathbf{r}_{\mathbf{s}_2 \mathbf{s}_2}(k) + \sigma^2 \delta(k), \quad (2.8)$$

$$\mathbf{r}_{\mathbf{x}_1 \mathbf{x}_2}(k) = h_{11} h_{21} \mathbf{r}_{\mathbf{s}_1 \mathbf{s}_1}(k) + h_{12} h_{22} \mathbf{r}_{\mathbf{s}_2 \mathbf{s}_2}(k) + \mathbf{r}_{\mathbf{n}_1 \mathbf{n}_2}(0) \delta(k), \quad (2.9)$$

Si le bruit est spatialement blanc alors $\mathbf{r}_{\mathbf{n}_1 \mathbf{n}_2}(0) = 0$

2.1.1 Résolution du système d'équations

Pour résoudre le système d'équations (2.7) à (2.9), nous proposons de définir les opérateurs suivants \mathbf{F}_{ij} , \mathbf{T}_{ij} , \mathbf{F}_i et \mathbf{T}_j tels que :

$$\mathbf{F}_1 = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_1}(k) \quad (2.10)$$

$$\mathbf{F}_2 = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{x}_2 \mathbf{x}_2}(k) \quad (2.11)$$

$$\mathbf{F}_{12} = \frac{1}{K} \sum_{k=1}^k \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_2}(k) \quad (2.12)$$

$$\mathbf{T}_1 = \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_1}(0) - \sigma^2 \quad (2.13)$$

$$\mathbf{T}_2 = \mathbf{r}_{\mathbf{x}_2 \mathbf{x}_2}(0) - \sigma^2 \quad (2.14)$$

$$\mathbf{T}_{12} = \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_2}(0) \quad (2.15)$$

et on pose :

$$\mathbf{R}_1 = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{s}_1 \mathbf{s}_1}(k) \quad (2.16)$$

$$\mathbf{R}_2 = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{s}_2 \mathbf{s}_2}(k) \quad (2.17)$$

L'hypothèse d'identifiabilité aveugle permet, sans perte de généralité, de supposer :
 $\mathbf{r}_{s_1 s_1}(0) = \mathbf{r}_{s_2 s_2}(0) = 1$.

En appliquant ces opérateurs aux équations (2.7) à (2.9), nous obtenons le système d'équations suivants :

$$\mathbf{F}_1 = h_{11}^2 \mathbf{R}_1 + h_{12}^2 \mathbf{R}_2, \quad (2.18)$$

$$\mathbf{T}_1 = h_{11}^2 + h_{12}^2, \quad (2.19)$$

$$\mathbf{F}_2 = h_{21}^2 \mathbf{R}_1 + h_{22}^2 \mathbf{R}_2, \quad (2.20)$$

$$\mathbf{T}_2 = h_{21}^2 + h_{22}^2, \quad (2.21)$$

$$\mathbf{F}_{12} = h_{11} h_{21} \mathbf{R}_1 + h_{12} h_{22} \mathbf{R}_2, \quad (2.22)$$

$$\mathbf{T}_{12} = h_{11} h_{21} + h_{12} h_{22}, \quad (2.23)$$

A partir des équations (2.18) et (2.19), nous obtenons :

$$h_{11}^2 = \frac{\mathbf{F}_1 - \mathbf{T}_1 \mathbf{R}_2}{\mathbf{R}_1 - \mathbf{R}_2}, \quad (2.24)$$

$$h_{12}^2 = \frac{\mathbf{F}_1 - \mathbf{T}_1 \mathbf{R}_1}{\mathbf{R}_2 - \mathbf{R}_1}, \quad (2.25)$$

Les équations (2.20) et (2.21) donnent :

$$h_{21}^2 = \frac{\mathbf{F}_2 - \mathbf{T}_2 \mathbf{R}_2}{\mathbf{R}_1 - \mathbf{R}_2}, \quad (2.26)$$

$$h_{22}^2 = \frac{\mathbf{F}_2 - \mathbf{T}_2 \mathbf{R}_1}{\mathbf{R}_2 - \mathbf{R}_1}, \quad (2.27)$$

A partir des équations (2.22) et (2.23), nous aurons :

$$h_{11} h_{21} = \frac{\mathbf{F}_{12} - \mathbf{T}_{12} \mathbf{R}_2}{\mathbf{R}_1 - \mathbf{R}_2}, \quad (2.28)$$

$$h_{12} h_{22} = \frac{\mathbf{F}_{12} - \mathbf{T}_{12} \mathbf{R}_1}{\mathbf{R}_2 - \mathbf{R}_1}, \quad (2.29)$$

Des équations (2.25) et (2.29), on obtient :

$$\frac{h_{12}}{h_{22}} = \frac{\mathbf{F}_1 - \mathbf{T}_1 \mathbf{R}_1}{\mathbf{F}_{12} - \mathbf{T}_{12} \mathbf{R}_1}, \quad (2.30)$$

Des équations (2.27) et (2.29), on obtient :

$$\frac{h_{22}}{h_{12}} = \frac{\mathbf{F}_2 - \mathbf{T}_2 \mathbf{R}_1}{\mathbf{F}_{12} - \mathbf{T}_{12} \mathbf{R}_1}, \quad (2.31)$$

En utilisant les équations (2.30) et (2.31), nous arrivons à l'équation du second degré suivante :

$$(\mathbf{F}_{12}^2 - \mathbf{F}_2\mathbf{F}_1) + (\mathbf{T}_2\mathbf{F}_1 + \mathbf{T}_1\mathbf{F}_2 - 2\mathbf{F}_{12}\mathbf{T}_{12})\mathbf{R}_1 + (\mathbf{T}_{12}^2 - \mathbf{T}_1\mathbf{T}_2)\mathbf{R}_1^2 = 0 \quad (2.32)$$

En suivant la même approche, des équations (2.18) et (2.28), on déduit :

$$\frac{h_{11}}{h_{21}} = \frac{\mathbf{F}_1 - \mathbf{T}_1\mathbf{R}_2}{\mathbf{F}_{12} - \mathbf{T}_{12}\mathbf{R}_2}, \quad (2.33)$$

Des équations (2.26) et (2.28), on obtient :

$$\frac{h_{21}}{h_{11}} = \frac{\mathbf{F}_2 - \mathbf{T}_2\mathbf{R}_2}{\mathbf{F}_{12} - \mathbf{T}_{12}\mathbf{R}_2}, \quad (2.34)$$

En utilisant les équations (2.33) et (2.34), on obtient l'équation du second degré suivante :

$$(\mathbf{F}_{12}^2 - \mathbf{F}_2\mathbf{F}_1) + (\mathbf{T}_2\mathbf{F}_1 + \mathbf{T}_1\mathbf{F}_2 - 2\mathbf{F}_{12}\mathbf{T}_{12})\mathbf{R}_2 + (\mathbf{T}_{12}^2 - \mathbf{T}_1\mathbf{T}_2)\mathbf{R}_2^2 = 0 \quad (2.35)$$

On remarque, dans les équations (2.32) et (2.35), que \mathbf{R}_1 et \mathbf{R}_2 sont solutions de la même équation du second ordre. Nous montrerons par la suite que cette équation possède deux solutions distinctes sous les conditions d'identifiabilité, $\mathbf{R}_1 \neq \mathbf{R}_2$.

A partir de la résolution de l'équation du second ordre (2.32) ou (2.35), on peut déduire les quantités \mathbf{R}_1 et \mathbf{R}_2 tels que :

$$\mathbf{R}_1 = \frac{b + c}{2a} \quad (2.36)$$

$$\mathbf{R}_2 = \frac{b - c}{2a} \quad (2.37)$$

sachant que :

$$b = 2\mathbf{F}_{12}\mathbf{F}_{12} - \mathbf{T}_2\mathbf{F}_1 + \mathbf{T}_1\mathbf{F}_2 \quad (2.38)$$

$$a = \mathbf{T}_{12}^2 - \mathbf{T}_1\mathbf{T}_2 \quad (2.39)$$

$$c^2 = (\mathbf{T}_2\mathbf{F}_1 - \mathbf{T}_1\mathbf{F}_2)^2 + 4(\mathbf{T}_{12}\mathbf{F}_2 - \mathbf{T}_2\mathbf{F}_{12})(\mathbf{T}_{12}\mathbf{F}_1 - \mathbf{T}_1\mathbf{F}_{12}) \quad (2.40)$$

On obtient alors :

$$h_{11}^2 = \frac{2a\mathbf{F}_1 - \mathbf{T}_1(b-c)}{2c} \quad (2.41)$$

$$h_{22}^2 = \frac{2a\mathbf{F}_2 - \mathbf{T}_2(b+c)}{2c} \quad (2.42)$$

$$h_{11}h_{21} = \frac{2a\mathbf{F}_{12} - \mathbf{T}_{12}(b-c)}{2c} \quad (2.43)$$

$$h_{12}h_{22} = \frac{2a\mathbf{F}_{12} - \mathbf{T}_{12}(b+c)}{2c} \quad (2.44)$$

Grâce à l'indétermination dû à l'identification aveugle, on remarque que :

$$\begin{pmatrix} 2c.h_{11}^2 & 2c.h_{22}h_{12} \\ 2c.h_{11}h_{21} & 2c.h_{22}^2 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} 2c.h_{11} & 0 \\ 0 & 2c.h_{22} \end{pmatrix}; \quad (2.45)$$

est aussi une solution. D'où une estimée de la matrice de mélange est donnée par :

$$\hat{\mathbf{H}} = \begin{pmatrix} 2a.\mathbf{F}_1 - \mathbf{T}_1(b-c) & 2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b+c) \\ 2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b-c) & 2a.\mathbf{F}_2 - \mathbf{T}_2(b+c) \end{pmatrix}; \quad (2.46)$$

2.1.2 Matrice de Séparation

À présent, l'objectif est de déterminer les coefficients de la matrice de séparation, notée :

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \quad (2.47)$$

Les signaux source seront estimés comme éléments du vecteur de $\mathbf{W} \mathbf{x}(t)$. Plusieurs critères de minimisation et de maximisation peuvent être utilisés pour optimiser les coefficients de pondération de la matrice de séparation [18]. Cela comprend la maximisation du rapport signal sur interférence plus bruit ou minimiser linéairement la variance des signaux source estimés. Ici, nous calculons le filtre spatial "zero-forcing", qui maximise le rapport signal sur interférence (SIR) à la sortie du filtre. Prenant en compte les indéterminations inhérentes à la séparation aveugle de sources, la solution du zero-forcing est donnée par :

$$\mathbf{W} \mathbf{H} = \mathbf{P} \mathbf{D} \quad (2.48)$$

Où \mathbf{P} et \mathbf{D} sont, respectivement, une matrice de permutation et une matrice diagonale. Une solution possible à l'équation (2.47) est proposée par :

$$\mathbf{W} = \begin{pmatrix} h_{22} & -h_{12} \\ -h_{21} & h_{11} \end{pmatrix} \quad (2.49)$$

Où les coefficients h_{ij} sont donnés par l'équation (2.46). La matrice de séparation est donnée alors par :

$$\widehat{\mathbf{W}} = \begin{pmatrix} 2a.\mathbf{F}_2 - \mathbf{T}_2(b+c) & -2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b+c) \\ -2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b-c) & 2a.\mathbf{F}_1 - \mathbf{T}_1(b-c) \end{pmatrix}; \quad (2.50)$$

Pour estimer les coefficients de la matrice de mélange, une estimation de la variance du bruit σ^2 est nécessaire. Elle peut être obtenue par la décomposition propre de la matrice de covariance de données si un troisième capteur est disponible [27]. Alternativement, σ^2 peut être estimée seulement à l'aide de deux capteurs avant le début de l'enregistrement des données. Il est également possible d'estimer σ^2 en choisissant la valeur qui minimise (au sens des moindres carrés) l'auto-corrélation (à des laps de temps différents) entre les deux sorties de $\mathbf{H}(\sigma^2)^{-1}\mathbf{x}(t)$. Notons qu'en pratique, les matrices de corrélation temporelle des données sont remplacées par leurs moyennes temporelles. Pour poursuivre une non-stationnarité éventuelle, ces matrices de corrélation temporelle peuvent être estimées de façon adaptative en utilisant, par exemple, une mémoire exponentielle. Cet aspect est traité au chapitre 4.

Dans cette section, nous avons formulé un nouvel algorithme analytique pour l'identification aveugle ASOBI simple et facilement implémentable dans le cas d'un bruit spatialement blanc avec une variance connue ou estimée. Ainsi, on présente l'algorithme ASOBI ayant la solution suivante :

$$\widehat{\mathbf{H}} = \begin{pmatrix} 2a.\mathbf{F}_1 - \mathbf{T}_1(b-c) & 2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b+c) \\ 2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b-c) & 2a.\mathbf{F}_2 - \mathbf{T}_2(b+c) \end{pmatrix}; \quad (2.51)$$

L'algorithme ASOBI est résumé dans le tableau Algorithm 1.

Algorithm 1 Résumé de l'algorithme ASOBI

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$;
 - 2: **Estimer** : $\mathbf{r}_{ij}(k)$, $i, j = 1, 2$ avec $k = 0, \dots, K$ en utilisant l'équation (2.6) ;
 - 3: **Calculer** : $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{12}, \mathbf{F}_1, \mathbf{F}_2$ et \mathbf{F}_{12} en utilisant les équations de (2.10) à (2.15) ;
 - 4: **Calculer** : b, a, c en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Estimer la matrice** : $\widehat{\mathbf{W}}$ selon l'équation (2.50) ;
 - 6: **Estimer les signaux source** : $\hat{\mathbf{s}}(t) = \widehat{\mathbf{W}}\mathbf{x}(t)$ ($t = 1, \dots, T$).
-

2.2 Algorithme Analytique pour l'identification aveugle k-ASOBI

Nous suivons ici la même démarche de la section précédente mais en choisissant d'autres opérateurs pour la résolution du système d'équations (2.7) à (2.9), ainsi :

$$\mathbf{F}'_1 = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_1}(k) \quad (2.52)$$

$$\mathbf{F}'_2 = \mathbf{r}_{\mathbf{x}_2\mathbf{x}_2}(k) \quad (2.53)$$

$$\mathbf{F}'_{12} = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_2}(k) \quad (2.54)$$

$$\mathbf{T}'_1 = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_1}(0) - \sigma^2 \quad (2.55)$$

$$\mathbf{T}'_2 = \mathbf{r}_{\mathbf{x}_2\mathbf{x}_2}(0) - \sigma^2 \quad (2.56)$$

$$\mathbf{T}'_{12} = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_2}(0) \quad (2.57)$$

on pose :

$$\mathbf{R}'_1 = \mathbf{r}_{\mathbf{s}_1\mathbf{s}_1}(k) \quad (2.58)$$

$$\mathbf{R}'_2 = \mathbf{r}_{\mathbf{s}_2\mathbf{s}_2}(k) \quad (2.59)$$

Nous obtenons la même solution que l'équation (2.46) mais avec des opérateurs différents. En partant toujours de l'hypothèse d'un bruit spatialement blanc de variance connue ou estimée. En choisissant un retard k arbitraire, on peut avoir des algorithmes tels que 1-ASOBI, 2-ASOBI, ..., k -ASOBI. On obtient alors la solution suivante :

$$\widehat{\mathbf{H}}' = \begin{pmatrix} 2a.\mathbf{F}'_1 - \mathbf{T}'_1(b-c) & 2a.\mathbf{F}'_{12} - \mathbf{T}'_{12}(b+c) \\ 2a.\mathbf{F}'_{12} - \mathbf{T}'_{12}(b-c) & 2a.\mathbf{F}'_2 - \mathbf{T}'_2(b+c) \end{pmatrix}; \quad (2.60)$$

et la matrice de séparation est donnée par :

$$\widehat{\mathbf{W}}' = \begin{pmatrix} 2a.\mathbf{F}'_2 - \mathbf{T}'_2(b+c) & -2a.\mathbf{F}'_{12} - \mathbf{T}'_{12}(b+c) \\ -2a.\mathbf{F}'_{12} - \mathbf{T}'_{12}(b-c) & 2a.\mathbf{F}'_1 - \mathbf{T}'_1(b-c) \end{pmatrix}; \quad (2.61)$$

L'algorithme, appelé k -ASOBI, est résumé dans le tableau Algorithm 2.

Algorithm 2 Résumé de l'algorithme k -ASOBI

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$;
 - 2: **Estimer** : $\mathbf{r}_{ij}(k)$, $i, j = 1, 2$ pour un choix arbitraire du retard k en utilisant l'équation (2.6) ;
 - 3: **Calculer** : $\mathbf{T}'_1, \mathbf{T}'_2, \mathbf{T}'_{12}, \mathbf{F}'_1, \mathbf{F}'_2$ et \mathbf{F}'_{12} en utilisant les équations de (2.52) à (2.57) ;
 - 4: **Calculer** : b, a, c en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Estimer la matrice** : $\widehat{\mathbf{W}}'$ selon l'équation (2.61) ;
 - 6: **Estimer les signaux source** : $\hat{\mathbf{s}}(t) = \widehat{\mathbf{W}}'\mathbf{x}(t)$ ($t = 1, \dots, T$).
-

2.3 Conditions d'identifiabilité

Dans cette section, nous déterminons les conditions qui permettent d'identifier la matrice \mathbf{H} du modèle (1.1) à une permutation et un facteur près de ses colonnes. Pour que la matrice \mathbf{H} soit identifiable, il est nécessaire et suffisant que l'équation (2.35) ait des solutions réelles et distinctes. Pour cela, il faudra que le discriminant de cette équation d'ordre 2 soit positif et non nul. Ce déterminant Δ est donné par :

$$\Delta = (\mathbf{T}_2\mathbf{F}_1 + \mathbf{T}_1\mathbf{F}_2 - 2\mathbf{F}_{12}\mathbf{T}_{12})^2 - 4((\mathbf{T}_{12})^2 - \mathbf{T}_1\mathbf{T}_2)((\mathbf{F}_{12})^2 - \mathbf{F}_2\mathbf{F}_1) \quad (2.62)$$

Le développement de cette équation nous donne :

$$\Delta = (\mathbf{T}_2\mathbf{F}_1 - \mathbf{T}_1\mathbf{F}_2)^2 + 4(\mathbf{F}_{12}\mathbf{F}_2 - \mathbf{T}_2\mathbf{F}_{12})(\mathbf{T}_{12}\mathbf{F}_1 - \mathbf{T}_1\mathbf{F}_{12}) \quad (2.63)$$

On remplace les expressions des termes \mathbf{F}_1 , \mathbf{F}_2 , \mathbf{F}_{12} , \mathbf{T}_1 , \mathbf{T}_2 et \mathbf{T}_{12} selon les équations (2.18) à (2.23)¹. On obtient après quelques calculs le résultat suivant :

$$\begin{aligned} \Delta = & [(h_{21}^2 + h_{22}^2)(h_{11}^2\mathbf{R}_1 + h_{12}^2\mathbf{R}_2) - (h_{11}^2 + h_{12}^2)(h_{21}^2\mathbf{R}_1 + h_{22}^2\mathbf{R}_2)]^2 + 4[(h_{11}h_{21} + h_{12}h_{22}) \times \\ & (h_{21}^2\mathbf{R}_1 + h_{22}^2\mathbf{R}_2) - (h_{21}^2 + h_{22}^2)(h_{11}h_{21}\mathbf{R}_1 + h_{12}h_{22}\mathbf{R}_2)] [(h_{11}h_{21} + h_{12}h_{22}) \times \\ & (h_{11}^2\mathbf{R}_1 + h_{12}^2\mathbf{R}_2) - (h_{21}^2 + h_{22}^2)(h_{11}h_{21}\mathbf{R}_1 + h_{12}h_{22}\mathbf{R}_2)] \end{aligned} \quad (2.64)$$

Alors :

$$\Delta = [(h_{11}h_{22} + h_{12}h_{21})(h_{11}h_{22} - h_{12}h_{21})(\mathbf{R}_1 - \mathbf{R}_2)]^2 + 4[-h_{22}h_{21}(h_{11}h_{22} - h_{12}h_{21}) \times (\mathbf{R}_1 - \mathbf{R}_2)h_{12}h_{11}(h_{11}h_{22} - h_{12}h_{21})(\mathbf{R}_1 - \mathbf{R}_2)] \quad (2.65)$$

Ainsi, on obtient :

$$\Delta = [(h_{11}h_{22} + h_{12}h_{21})(h_{11}h_{22} - h_{12}h_{21})(\mathbf{R}_1 - \mathbf{R}_2)]^2 - 4h_{22}h_{21}h_{12}h_{11}[(h_{11}h_{22} - h_{12}h_{21}) \times (\mathbf{R}_1 - \mathbf{R}_2)]^2 \quad (2.66)$$

On obtient finalement le résultat suivant :

$$\Delta = (h_{11}h_{22} - h_{12}h_{21})^4(\mathbf{R}_1 - \mathbf{R}_2)^2 \quad (2.67)$$

Alors :

$$\Delta = (\det(\mathbf{H}))^4(\mathbf{R}_1 - \mathbf{R}_2)^2 \quad (2.68)$$

Il est clair que la matrice \mathbf{H} est identifiable si et seulement si $\det(\mathbf{H}) \neq 0$ et $\mathbf{R}_1 \neq \mathbf{R}_2$:

1. On aura exactement les mêmes expressions aussi bien pour les algorithmes ASOBI et k-ASOBI.

- La première condition est toujours vérifiée puisque dans le modèle (1.1) la matrice \mathbf{H} est supposée de rang plein.
- La deuxième condition va dépendre des statistiques utilisés pour chaque algorithme pour rendre $\mathbf{R}_1 \neq \mathbf{R}_2$.

Il est clair que si les sources ont des spectres normalisés identiques, \mathbf{R}_1 sera égal à \mathbf{R}_2 quelque soit la statistique d'ordre deux choisie. Dans ce cas, les algorithmes proposés ne pourront pas séparer des signaux sources avec des spectres normalisés identique.

2.4 Algorithme Analytique pour l'identification aveugle Optimal k-ASOBI

Dans cette section, nous exploitons les résultats d'identification trouvés dans la section précédente, ainsi nous nous intéressons à l'algorithme k-ASOBI dans le cas où le choix du retard k rend l'algorithme k-ASOBI le plus performant possible. En effet, le choix du retard k est sélectionné d'une manière optimale, ainsi c'est la valeur de k qui rend la différence $(\mathbf{R}_1'' - \mathbf{R}_2'')$ maximale. Comme la valeur ² de $(\mathbf{c}^2(k))$ est proportionnel à $(\mathbf{R}_1'' - \mathbf{R}_2'')$ (voir section 2.3) :

$$\mathbf{c}^2(k) = (\mathbf{T}_2''\mathbf{F}_1'' - \mathbf{T}_1''\mathbf{F}_2'')^2 + 4(\mathbf{T}_{12}''\mathbf{F}_2'' - \mathbf{T}_2''\mathbf{F}_{12}'')(\mathbf{T}_{12}''\mathbf{F}_1'' - \mathbf{T}_1''\mathbf{F}_{12}'') = \det(\mathbf{H})^4(\mathbf{R}_1'' - \mathbf{R}_2'')^2 \quad (2.69)$$

on aura donc :

$$\mathbf{k}\text{-optimal} = \operatorname{argmax}_k(\mathbf{c}^2(k)) \quad (2.70)$$

Si pour toutes les valeurs de k, $\mathbf{c}^2(k)$ est nulle (ou égale au zéro machine), ceci nous renseigne que le problème de séparation de sources n'est pas solvable au second ordre.

Une fois la valeur optimale de k est trouvée, on obtient alors la solution suivante

$$\widehat{\mathbf{H}}'' = \begin{pmatrix} 2a.\mathbf{F}_1'' - \mathbf{T}_1''(b - c) & 2a.\mathbf{F}_{12}'' - \mathbf{T}_{12}''(b + c) \\ 2a.\mathbf{F}_{12}'' - \mathbf{T}_{12}''(b - c) & 2a.\mathbf{F}_2'' - \mathbf{T}_2''(b + c) \end{pmatrix}; \quad (2.71)$$

et la matrice de séparation est donnée par :

$$\widehat{\mathbf{W}}'' = \begin{pmatrix} 2a.\mathbf{F}_2'' - \mathbf{T}_2''(b + c) & -2a.\mathbf{F}_{12}'' - \mathbf{T}_{12}''(b + c) \\ -2a.\mathbf{F}_{12}'' - \mathbf{T}_{12}''(b - c) & 2a.\mathbf{F}_1'' - \mathbf{T}_1''(b - c) \end{pmatrix}; \quad (2.72)$$

L'algorithme, appelé Optimal k-ASOBI, est résumé dans la tableau Algorithm 3.

2. $(\mathbf{c}^2(k))$ représente le discriminant représenté par l'équation (2.62).

Algorithm 3 Résumé de l'algorithme Optimal K-ASOBI

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$;
 - 2: **Calculer** : la valeur optimale de k en utilisant les équations (2.69) et (2.70) ;
 - 3: **Estimer** : $\mathbf{r}_{ij}(k)$, $i, j = 1, 2$ pour un ce choix optimal du retard k ;
 - 4: **Calculer** : $\mathbf{T}_1'', \mathbf{T}_2'', \mathbf{T}_{12}'', \mathbf{F}_1'', \mathbf{F}_2''$ et \mathbf{F}_{12}'' en utilisant les équations de (2.52) à (2.57) ;
 - 5: **Calculer** : b, a, c en utilisant les équations de (2.38) à (2.40) ;
 - 6: **Estimer la matrice** : $\hat{\mathbf{W}}''$ selon l'équation (2.72) ;
 - 7: **Estimer les signaux source** : $\hat{\mathbf{s}}(t) = \hat{\mathbf{W}}'' \mathbf{x}(t)$ ($t = 1, \dots, T$).
-

2.5 Simulation et Évaluation des performances

Afin de vérifier l'efficacité de l'algorithme ASOBI et ses deux versions, nous avons pris deux signaux de parole échantillonnés à 8000Hz que nous avons mélangé par une matrice aléatoire :

Les observations de la figure 2.1 montrent les simulations des deux signaux de parole, leurs mélanges par la matrice \mathbf{H} et leur estimation par les algorithmes ASOBI, k-ASOBI et k-ASOBI Optimal. Il est clair que ces algorithmes donnent de bons résultats pour cet exemple. Nous obtenons finalement une bonne séparation à un coefficient et une permutation près.

De même pour la Figure 2.2 où deux signaux EEG et ECG sont utilisés. Par conséquent, les solutions proposées fonctionnent correctement dans ce cas et donnent des résultats similaires.

2.5.1 Critère de performance

La définition de critères d'évaluation est une étape importante pour comparer différents algorithmes de séparation de sources. Il paraît logique d'utiliser des critères du même type que ceux utilisés en débruitage, c'est-à-dire le plus souvent l'évaluation de rapports signal à bruit. Cependant, nous nous trouvons dans une situation légèrement différente du débruitage, car nous avons la possibilité de prendre plusieurs signaux de référence, à savoir les différentes sources de départ, alors que dans le cas du débruitage, seul le signal original importe [39].

Un critère objectif est fondé sur l'interférence entre la source originale recherchée et l'autre source originale que l'on voulait atténuer, voire éliminer. Nous appellerons ce critère SIR pour rapport source sur interférence (Source-to-Interference Ratio).

Après l'identification aveugle, les signaux sources estimés sont donnés par :

$$\hat{\mathbf{s}}(t) = \hat{\mathbf{W}} \mathbf{x}(t) = \hat{\mathbf{W}}\mathbf{H} \mathbf{s}(t) + \hat{\mathbf{W}} \mathbf{n}(t) \quad (2.73)$$

Où $\hat{\mathbf{W}}$ est une estimation de la matrice du filtre spatial. La matrice $\hat{\mathbf{P}}$ définie par : $\hat{\mathbf{P}} = \hat{\mathbf{W}}\mathbf{H}$ doit être proche d'une certaine matrice de permutation multipliée par une

2.5. SIMULATION ET ÉVALUATION DES PERFORMANCES

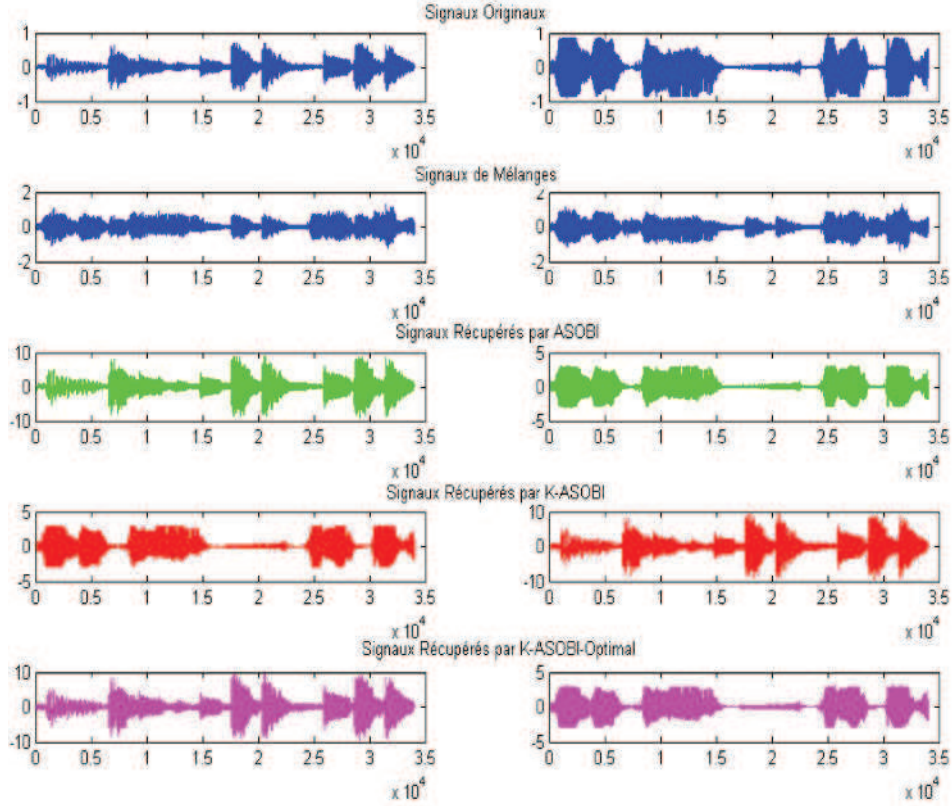


Figure 2.1 — Résultat de la simulation : signaux de parole; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l’algorithme ASOBI , Ligne 4 : Signaux Récupérés par l’algorithme k-ASOBI, Ligne 5 : Signaux Récupérés par l’algorithme Optimal k-ASOBI

matrice diagonale (permutation et indéterminations d’échelle). La p^{eme} source estimée est :

$$\hat{s}_p(t) = \sum_{q=1}^2 \hat{\mathbf{P}}_{pq} \hat{s}_q(t) \quad (2.74)$$

et contient le q^{eme} signal source avec un taux de :

$$\frac{\|\hat{\mathbf{P}}_{pq}\|^2}{\|\hat{\mathbf{P}}_{pp}\|^2} \quad (2.75)$$

Suite à cela, nous présentons quelques simulations pour évaluer les performances de la solution par le critère du taux de réjection moyen défini par :

$$I_{perf} = \sum_{q \neq p} \frac{\|\hat{\mathbf{P}}_{pq}\|^2}{\|\hat{\mathbf{P}}_{pp}\|^2} \quad (2.76)$$

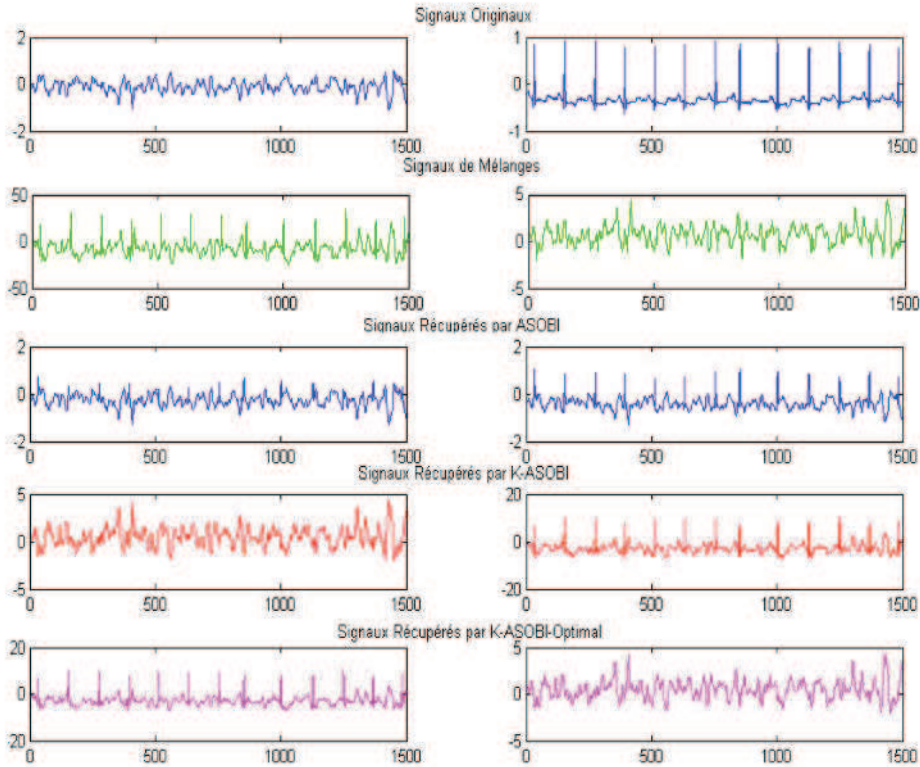


Figure 2.2 — Résultat de la simulation : signaux EEG et ECG ; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l’algorithme ASOBI, Ligne 4 : Signaux Récupérés par l’algorithme k-ASOBI , Ligne 5 : Signaux Récupérés par l’algorithme k-ASOBI Optimal

2.5.2 Performance de l’algorithme ASOBI

Considérons deux sources mélangées par une matrice aléatoire \mathbf{H} . Le bruit additif est un processus gaussien de moyenne nulle et de matrice de covariance $\sigma^2 \mathbf{I}_N$. Les signaux sources ont une variance unité et chacun est généré par le filtrage d’un processus gaussien blanc à l’aide de deux filtres autorégressifs différents. Le taux de réjection global est moyenné sur 100 réalisations. Dans ces simulations, les solutions proposées supposent que le bruit est absent (c’est-à-dire dans les équations 2.13 à 2.15, σ^2 est mise à zéro) et $k = 8$. Sur la figure 2.3, le taux de réjection I_{perf} est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille 1024 . Dans la figure 2.4, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l’échantillon. Les figures 2.3 et 2.4 indiquent une amélioration significative des performances pour un nombre suffisamment grand d’échantillons et un SNR élevé.

2.5.3 Performance de l’algorithme k-ASOBI

Des simulations similaires sont effectuées dans les mêmes conditions que la section 2.5.2, en traçant des courbes de performances de l’algorithme k-ASOBI pour plusieurs

2.5. SIMULATION ET ÉVALUATION DES PERFORMANCES

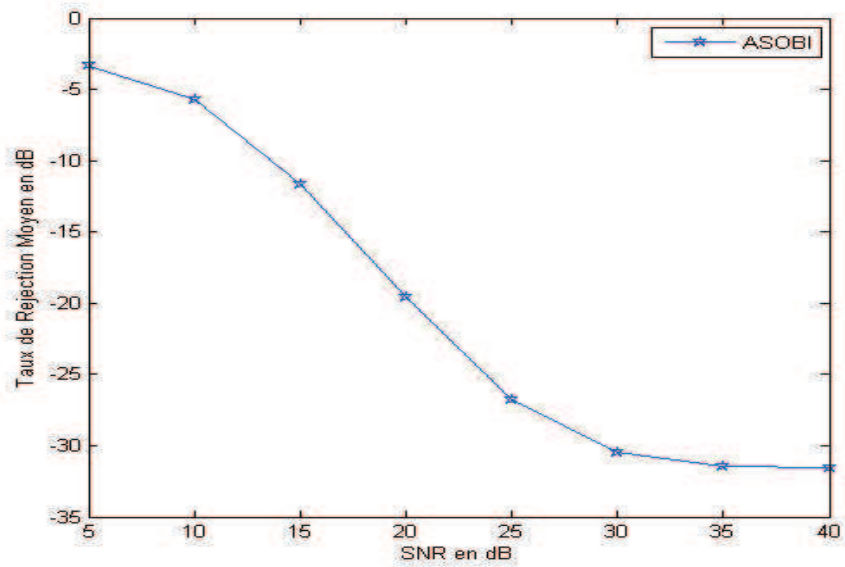


Figure 2.3 — Taux de réjection moyen vs SNR. ASOBI

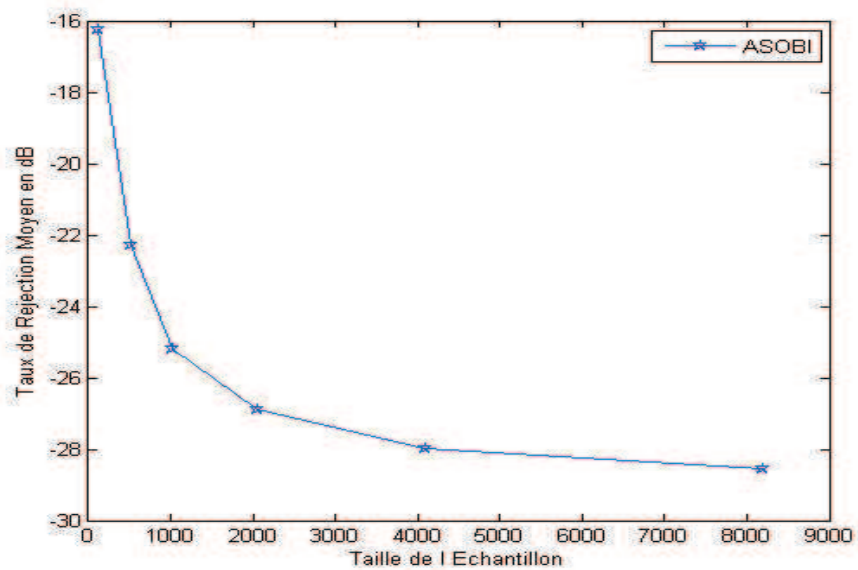


Figure 2.4 — Taux de réjection moyen vs Taille de l'échantillon. ASOBI

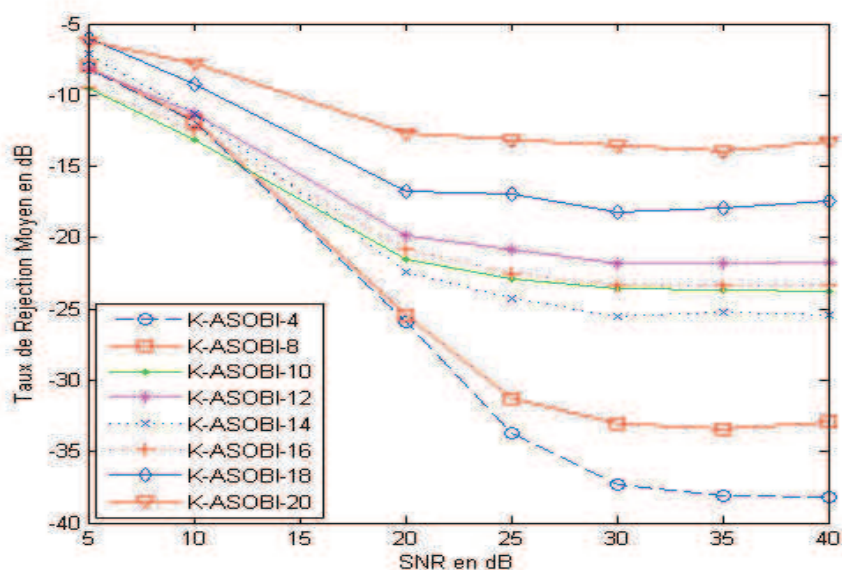


Figure 2.5 — Taux de réjection moyen vs SNR. K-ASOBI

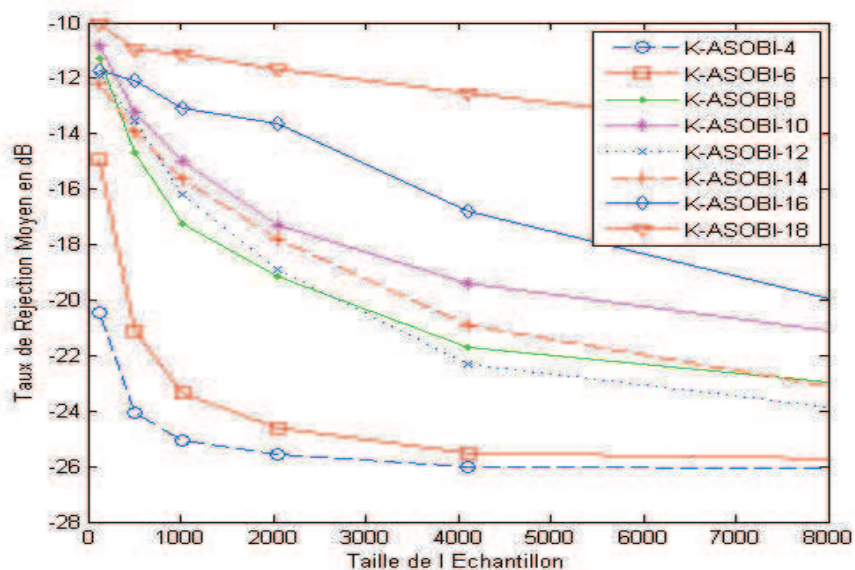


Figure 2.6 — Taux de réjection moyen vs Taille de l'échantillon. K-ASOBI

valeurs de k . En effet, sur la figure 2.5, le taux de réjection I_{perf} est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille 1024.

Dans la figure 2.6, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l'échantillon. Les figures 2.5 et 2.6 indiquent une amélioration significative des performances pour un nombre suffisamment élevé d'échantillons et un SNR élevé.

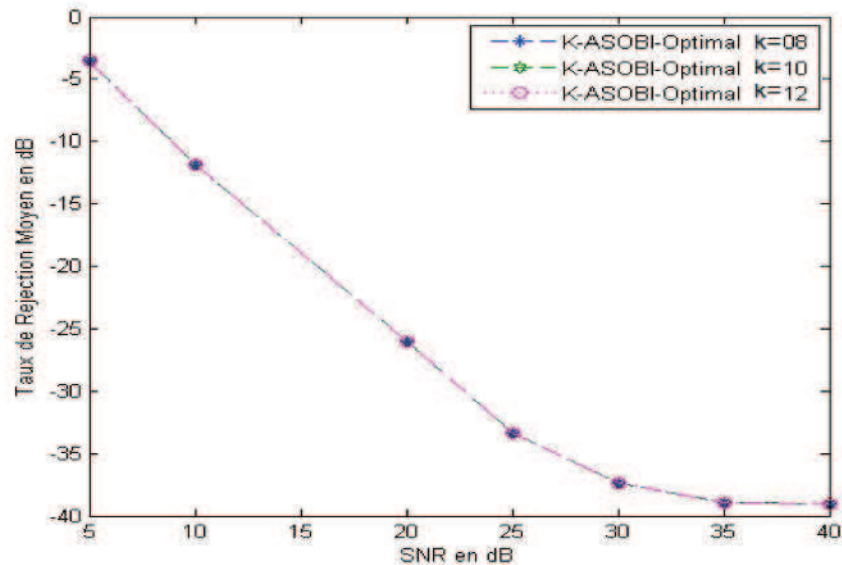


Figure 2.7 — Taux de réjection moyen vs SNR. Optimal K-ASOBI

2.5.4 Performance de l'algorithme Optimal k-ASOBI

Des simulations similaires sont effectuées dans les mêmes conditions que la section (2.5.2), en traçant des courbes de performances de l'algorithme Optimal k-ASOBI. En effet, sur la figure 2.7, le taux de réjection I_{perf} est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille 1024 et une plage de retard de 1 à 8, puis de 1 à 10 ensuite de 1 à 12.

Dans la figure 2.8, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l'échantillon. Les figures 2.7 et 2.8 indiquent une amélioration significative des performances pour un nombre suffisamment grand d'échantillons et un SNR élevé. Ainsi il est remarqué que les trois courbes se confondent car, à chaque fois, l'algorithme Optimal k-ASOBI calcule la valeur de k qui permet d'obtenir les meilleures performances.

Une comparaison est effectuée entre les trois algorithmes (ASOBI, k-ASOBI et Optimal k-ASOBI) pour $k=8$. En effet, les figures 2.9 et 2.10 montrent que l'algorithme Optimal k-ASOBI présente les meilleures performances.

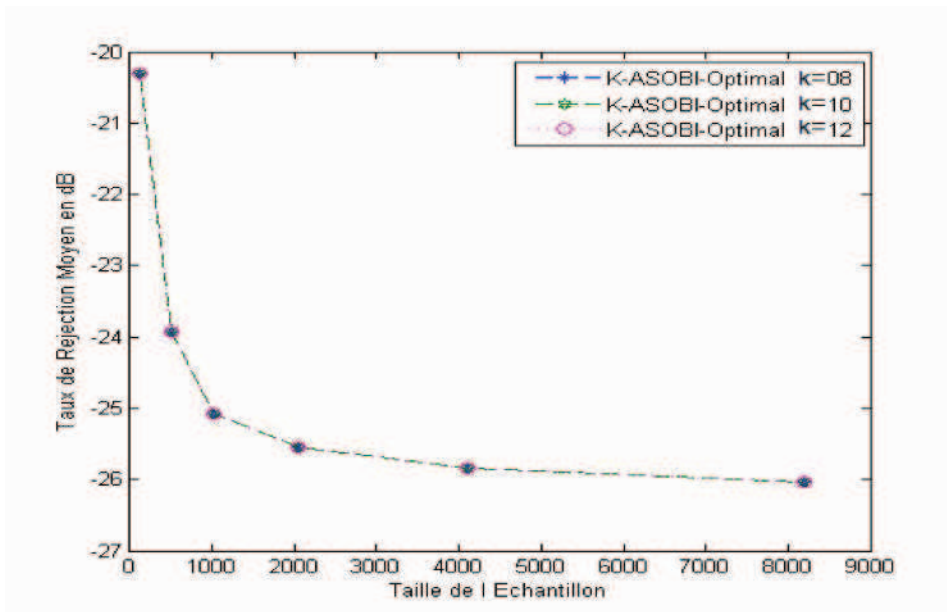


Figure 2.8 — Taux de réjection moyen vs Taille de l'échantillon. Optimal K-ASOBI

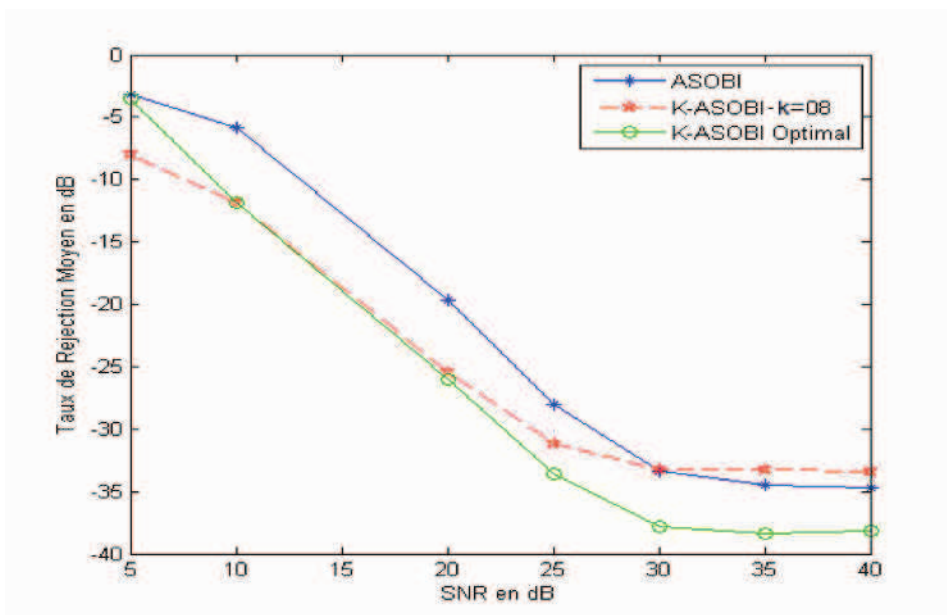


Figure 2.9 — Taux de réjection moyen vs SNR. Comparaison des trois algorithmes

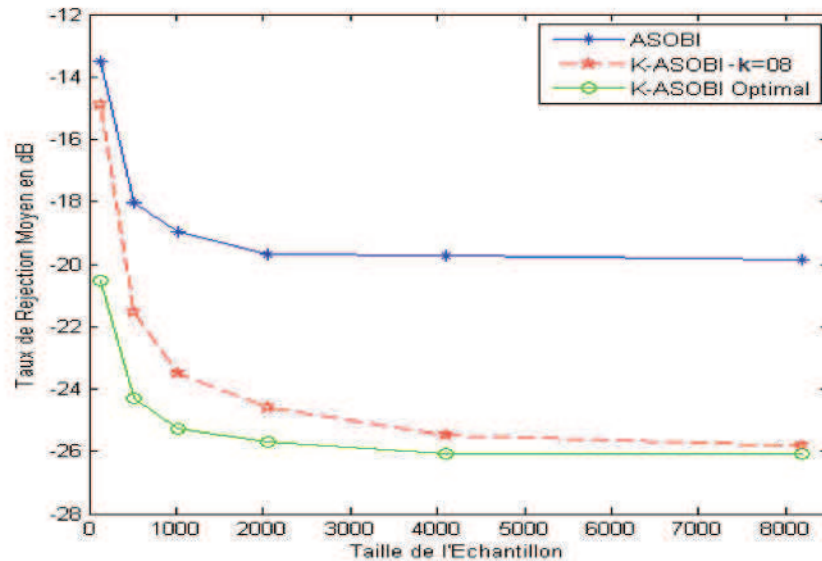


Figure 2.10 — Taux de réjection moyen vs Taille de l'échantillon. comparaison des trois algorithmes

2.6 Conclusion

Dans ce chapitre, nous avons proposé une première classe d'algorithmes pour le cas de bruit additif spatialement blanc. Ainsi, nous avons développé les algorithmes ASOBI, k-ASOBI et Optimal k-ASOBI. L'identifiabilité des algorithmes proposés a été étudiée et les conditions pour lesquelles les coefficients de la matrice de mélange peuvent être identifiables ont été déduites. Celles-ci sont fonction des statistiques utilisées par chaque algorithme proposé. Si les signaux sources présentent des spectres normalisés identiques, les algorithmes proposés ne permettent pas de les séparer. Il faudra alors avoir recours aux statistiques d'ordre supérieur à deux.

La particularité des algorithmes développés est leur faible coût de calcul. Ces algorithmes ne nécessitent pas d'opérations de division, ce qui peut rendre leur implémentation hardware plus aisées. Ces opérations, dont l'implémentation est onéreuse sont évitées grâce à l'exploitation du concept de l'identification aveugle qui stipule que le recouvrement des signaux sources ne peut se faire qu'à un facteur et une permutation près. Un avantage majeur des algorithmes proposés est le fait qu'ils fournissent les estimés de la matrice de mélange et des signaux sources par un calcul direct sans avoir recours à aucune procédure itérative et donc évitant tout problème de convergence. Les simulations ont montré que les algorithmes développés présentent de bonnes performances.

Résolution analytique du problème de l'identification aveugle en présence de Bruit spatialement Coloré

Dans le chapitre précédent nous avons constaté qu'une évaluation de la variance du bruit est nécessaire pour une estimation robuste des coefficients du canal. Elle peut être obtenue par plusieurs méthodes : décomposition en valeurs propres de la matrice de covariance des mélanges si un troisième capteur est disponible. Autrement, elle peut être estimée à partir de deux capteurs avant enregistrement des données. Dans ce chapitre nous nous intéressons à de nouveaux algorithmes dont on n'a pas besoin d'estimer le bruit ; celui-ci, dans ce cas peut être spatialement coloré.

3.1 Algorithme Analytique pour l'identification aveugle Robuste ASOBI

Pour ne pas prendre en compte la contribution du bruit, on reprend la même solution de la section 2.1.1 avec de nouveaux opérateurs qu'on note \mathbf{F}_{ij}^r , \mathbf{T}_{ij}^r , \mathbf{F}_i^r et \mathbf{T}_j^r :

$$\mathbf{F}_1^r = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_1}(k) \quad (3.1)$$

$$\mathbf{F}_2^r = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{x}_2 \mathbf{x}_2}(k) \quad (3.2)$$

$$\mathbf{F}_{12}^r = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_2}(k) \quad (3.3)$$

$$\mathbf{T}_1^r = \mathbf{r}_{\mathbf{x}_1 \mathbf{x}_1}(1) \quad (3.4)$$

$$\mathbf{T}_2^r = \mathbf{r}_{\mathbf{x}_2\mathbf{x}_2}(1) \quad (3.5)$$

$$\mathbf{T}_{12}^r = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_2}(1) \quad (3.6)$$

on pose :

$$\mathbf{R}_1^r = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{s}_1\mathbf{s}_1}(k) \quad (3.7)$$

$$\mathbf{R}_2^r = \frac{1}{K} \sum_{k=1}^K \mathbf{r}_{\mathbf{s}_2\mathbf{s}_2}(k) \quad (3.8)$$

L'hypothèse d'identifiabilité aveugle permet de supposer, sans perte de généralité, que : $\mathbf{r}_{\mathbf{s}_1\mathbf{s}_1}(1) = \mathbf{r}_{\mathbf{s}_2\mathbf{s}_2}(1) = 1$. On remarque que la connaissance de la puissance du bruit n'est pas nécessaire.

En suivant la même démarche que dans le chapitre précédent, la matrice de séparation est donnée alors par :

$$\widehat{\mathbf{W}}^r = \begin{pmatrix} 2a.\mathbf{F}_2^r - \mathbf{T}_2^r(b+c) & -2a.\mathbf{F}_{12}^r - \mathbf{T}_{12}^r(b+c) \\ -2a.\mathbf{F}_{12}^r - \mathbf{T}_{12}^r(b-c) & 2a.\mathbf{F}_1^r - \mathbf{T}_1^r(b-c) \end{pmatrix}; \quad (3.9)$$

L'algorithme, appelé Robust ASOBI, est résumé dans le tableau Algorithm 4.

Algorithm 4 Résumé de l'algorithme Robust ASOBI

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$;
 - 2: **Estimer** : $\mathbf{r}_{ij}(k)$, $i, j = 1, 2$ avec $k = 0, \dots, K$ en utilisant l'équation (2.6) ;
 - 3: **Calculer** : $\mathbf{T}_1^r, \mathbf{T}_2^r, \mathbf{T}_{12}^r, \mathbf{F}_1^r, \mathbf{F}_2^r$ et \mathbf{F}_{12}^r en utilisant les équations de (3.1) à (3.6) ;
 - 4: **Calculer** : b, a, c en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Estimer la matrice** : $\widehat{\mathbf{W}}^r$ selon l'équation (3.9) ;
 - 6: **Estimer les signaux source** : $\hat{\mathbf{s}}(t) = \widehat{\mathbf{W}}^r \mathbf{x}(t)$ ($t = 1, \dots, T$).
-

3.2 Algorithme Analytique pour l'identification aveugle Robuste k-ASOBI

Dans cette partie, nous choisissons un retard k arbitraire, nous pouvons ainsi avoir Robust-1-ASOBI, Robust-2-ASOBI, ..., Robust- k -ASOBI, on définit alors :

$$\mathbf{F}_1^{r'} = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_1}(k) \quad (3.10)$$

$$\mathbf{F}_2^{r'} = \mathbf{r}_{\mathbf{x}_2\mathbf{x}_2}(k) \quad (3.11)$$

$$\mathbf{F}_{12}^{r'} = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_2}(k) \quad (3.12)$$

$$\mathbf{T}_1^{r'} = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_1}(1) \quad (3.13)$$

$$\mathbf{T}_2^{r'} = \mathbf{r}_{\mathbf{x}_2\mathbf{x}_2}(1) \quad (3.14)$$

$$\mathbf{T}_{12}^{r'} = \mathbf{r}_{\mathbf{x}_1\mathbf{x}_2}(1) \quad (3.15)$$

on pose

$$\mathbf{R}_1^{r'} = \mathbf{r}_{\mathbf{s}_1\mathbf{s}_1}(k) \quad (3.16)$$

$$\mathbf{R}_2^{r'} = \mathbf{r}_{\mathbf{s}_2\mathbf{s}_2}(k) \quad (3.17)$$

de même que précédemment, on obtient la solution :

$$\widehat{\mathbf{W}}^{r'} = \begin{pmatrix} 2a.\mathbf{F}_2^{r'} - \mathbf{T}_2^{r'}(b+c) & -2a.\mathbf{F}_{12}^{r'} - \mathbf{T}_{12}^{r'}(b+c) \\ -2a.\mathbf{F}_{12}^{r'} - \mathbf{T}_{12}^{r'}(b-c) & 2a.\mathbf{F}_1^{r'} - \mathbf{T}_1^{r'}(b-c) \end{pmatrix}; \quad (3.18)$$

L'algorithme, appelé Robuste k-ASOBI, est résumé dans le tableau Algorithm 5.

Algorithm 5 Résumé de l'algorithme Robust k-ASOBI

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$;
 - 2: **Estimer** : $\mathbf{r}_{ij}^{r'}(k)$, pour un choix arbitraire du retard k en utilisant l'équation (2.6) ;
 - 3: **Calculer** : $\mathbf{T}_1^{r'}, \mathbf{T}_2^{r'}, \mathbf{T}_{12}^{r'}, \mathbf{F}_1^{r'}, \mathbf{F}_2^{r'}$ et $\mathbf{F}_{12}^{r'}$ en utilisant les équations de (3.10) à (3.15) ;
 - 4: **Calculer** : b, a, c en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Estimer la matrice** : $\widehat{\mathbf{W}}^{r'}$ selon équation (3.18) ;
 - 6: **Estimer les signaux source** : $\hat{\mathbf{s}}(t) = \widehat{\mathbf{W}}^{r'} \mathbf{x}(t)$ ($t = 1, \dots, T$).
-

3.3 Algorithme Analytique pour l'identification aveugle Robuste Optimal k-ASOBI

L'algorithme proposé ici est identique à l'algorithme Robust-k-ASOBI, à l'exception du choix du retard k qui est sélectionné d'une manière optimale, c'est la valeur de k qui rend la différence $(\mathbf{R}_1^{r''} - \mathbf{R}_2^{r''})$ maximale, comme $(\mathbf{c}^2(k))$ est proportionnel à $(\mathbf{R}_1^{r''} - \mathbf{R}_2^{r''})^2$ (voir l'équation 2.68) on a :

$$\mathbf{c}^2(k) = (\mathbf{T}_2^{r''} \mathbf{F}_1^{r''} - \mathbf{T}_1^{r''} \mathbf{F}_2^{r''})^2 + 4(\mathbf{T}_{12}^{r''} \mathbf{F}_2^{r''} - \mathbf{T}_2^{r''} \mathbf{F}_{12}^{r''})(\mathbf{T}_{12}^{r''} \mathbf{F}_1^{r''} - \mathbf{T}_1^{r''} \mathbf{F}_{12}^{r''}) = \det(\mathbf{H})^4 (\mathbf{R}_1^{r''} - \mathbf{R}_2^{r''})^2 \quad (3.19)$$

on aura donc :

$$\mathbf{k}\text{-optimal} = \operatorname{argmax}_k(\mathbf{c}^2(k)) \quad (3.20)$$

On obtient alors la solution suivante :

$$\widehat{\mathbf{W}}^{r''} = \begin{pmatrix} 2a.\mathbf{F}_2^{r''} - \mathbf{T}_2^{r''}(b+c) & -2a.\mathbf{F}_{12}^{r''} - \mathbf{T}_{12}^{r''}(b+c) \\ -2a.\mathbf{F}_{12}^{r''} - \mathbf{T}_{12}^{r''}(b-c) & 2a.\mathbf{F}_1^{r''} - \mathbf{T}_1^{r''}(b-c) \end{pmatrix}; \quad (3.21)$$

Algorithm 6 Résumé de l'algorithme Robust k-ASOBI Optimal

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$;
 - 2: **Calculer** : la valeur optimale de k en utilisant les équations (3.19) et (3.20) ;
 - 3: **Estimer** : $\mathbf{r}_{ij}(k)^{r''}$, pour un choix optimal du retard k en utilisant les équations (2.58) et (2.59) ;
 - 4: **Calculer** : $\mathbf{T}_1^{r''}, \mathbf{T}_2^{r''}, \mathbf{T}_{12}^{r''}, \mathbf{F}_1^{r''}, \mathbf{F}_2^{r''}$ et $\mathbf{F}_{12}^{r''}$ en utilisant les équations de (2.52) à (2.57) ;
 - 5: **Calculer** : b, a, c en utilisant les équations de (2.38) à (2.40) ;
 - 6: **Estimer la matrice** : $\hat{\mathbf{W}}^{r''}$ selon équation (3.21) sachant que $\hat{\mathbf{W}}^{r''} = (\hat{\mathbf{H}}^{r''})^{-1}$ à un coefficient près ;
 - 7: **Estimer les signaux source** : $\hat{\mathbf{s}}(t) = \hat{\mathbf{W}}^{r''} \mathbf{x}(t)$ ($t = 1, \dots, T$).
-

L'algorithme, appelé Robust k-ASOBI Optimal, est résumé dans le tableau Algorithm 6.

Les conditions d'identifiabilité pour l'algorithme Robust ASOBI et ses versions sont identiques à ceux déjà données à la section (2.3).

3.4 Simulation et Évaluation des performances

Afin de vérifier l'efficacité des algorithmes Robuste ASOBI, Robuste k-ASOBI et Robuste k-ASOBI Optimal, nous avons pris deux signaux de parole échantillonnés à 8000Hz que nous avons mélangé par une matrice aléatoire (comme à la section 3.4).

Les observations de la figure 3.1 montrent les simulations des deux signaux de parole, leurs mélanges par la matrice \mathbf{H} et leur estimation par les algorithmes. Il devient clair que ces algorithmes donnent de bons résultats pour cet exemple. Nous obtenons finalement une bonne séparation à un coefficient et une permutation près. De même pour la Figure 3.2 où deux signaux EEG et ECG sont utilisés. Ainsi, les solutions proposées fonctionnent bien dans ce cas et donnent des résultats similaires.

3.4.1 Performance de l'algorithme Robust ASOBI

Les simulations sont réalisées dans les mêmes conditions de la section 3.4. Les solutions proposées utilisent les équations où le bruit n'est pas représenté ; ainsi on n'a pas besoin de l'estimer contrairement à l'algorithme ASOBI. Sur la figure 3.3, le taux de réjection I_{perf} (pour Robust ASOBI et ASOBI) est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille 1024 .

Dans la figure 3.4, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l'échantillon. Les figures 3.3 et 3.4 indiquent une amélioration significative des performances pour un nombre suffisamment grand d'échantillons et un SNR élevé. Il est bien évident que l'algorithme Robust ASOBI donne de bien meilleurs résultats que ceux de ASOBI.

3.4. SIMULATION ET ÉVALUATION DES PERFORMANCES

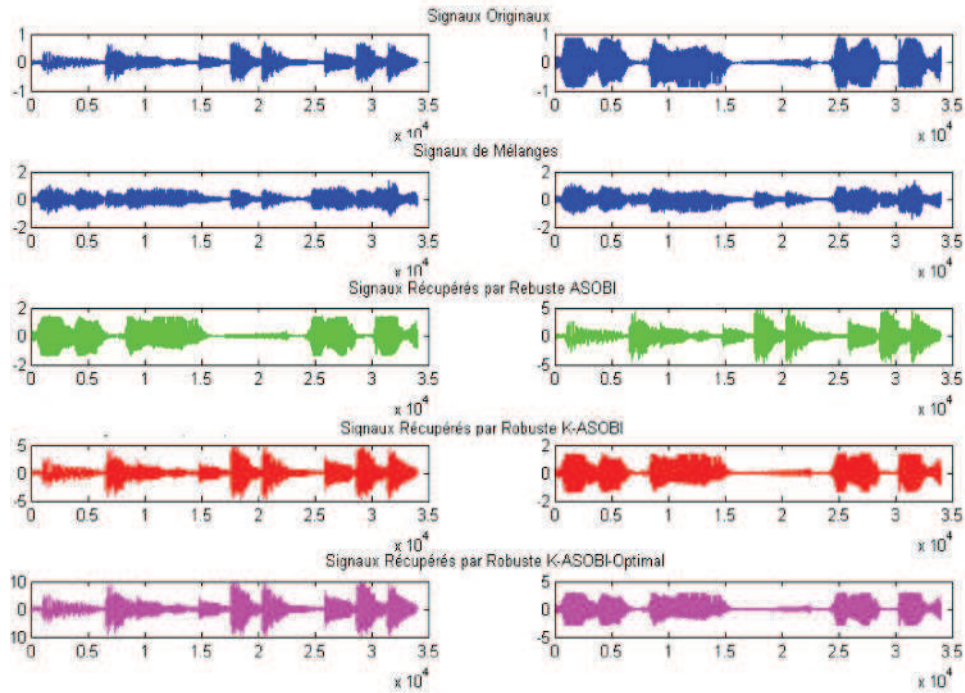


Figure 3.1 — Résultat de la simulation : signaux de parole ; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l’algorithme Robuste ASOBI , Ligne 4 : Signaux Récupérés par l’algorithme Robuste k-ASOBI, Ligne 5 : Signaux Récupérés par l’algorithme Robuste Optimal k-ASOBI

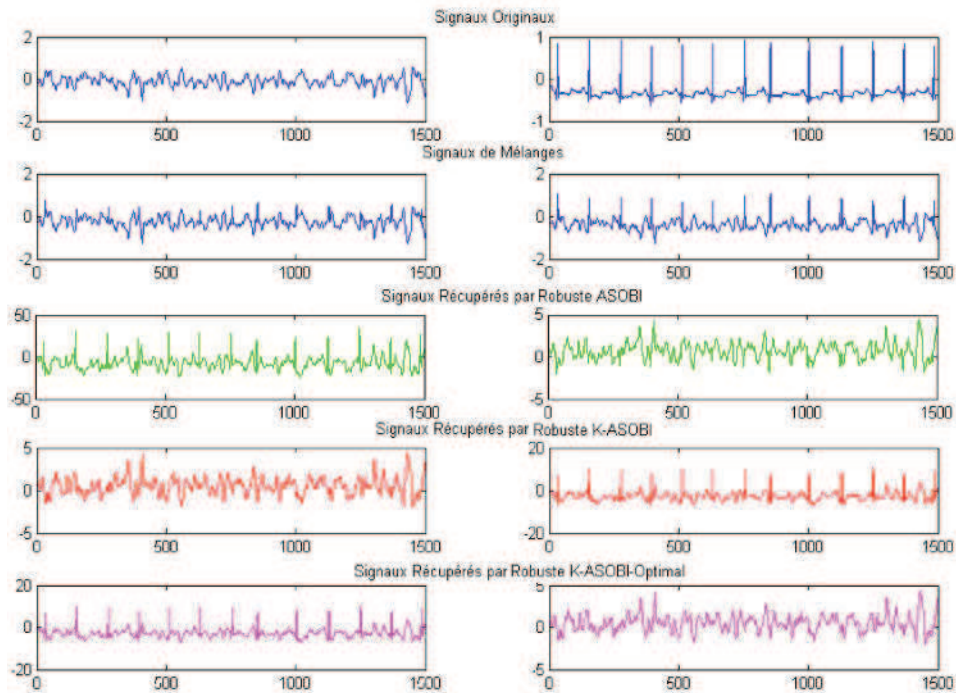


Figure 3.2 — Résultat de la simulation : signaux EEG et ECG ; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l’algorithme Robuste ASOBI, Ligne 4 : Signaux Récupérés par l’algorithme Robuste k-ASOBI , Ligne 5 : Signaux Récupérés par l’algorithme Robuste k-ASOBI Optimal

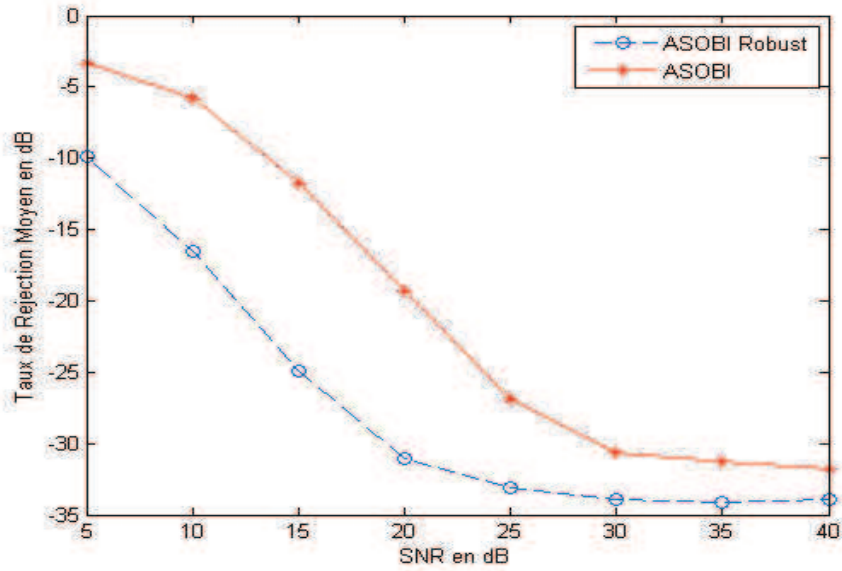


Figure 3.3 — Taux de réjection moyen vs SNR. Robust ASOBI

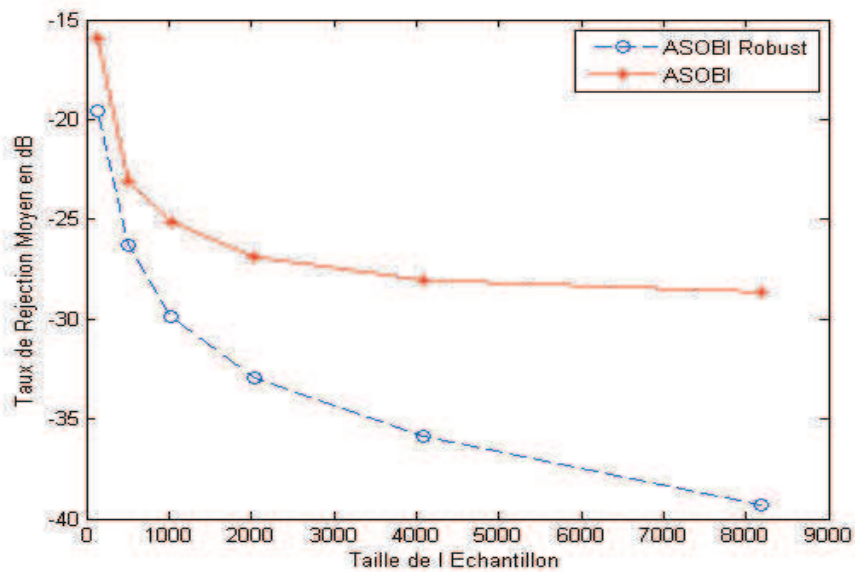


Figure 3.4 — Taux de réjection moyen vs Taille de l'échantillon. Robust ASOBI

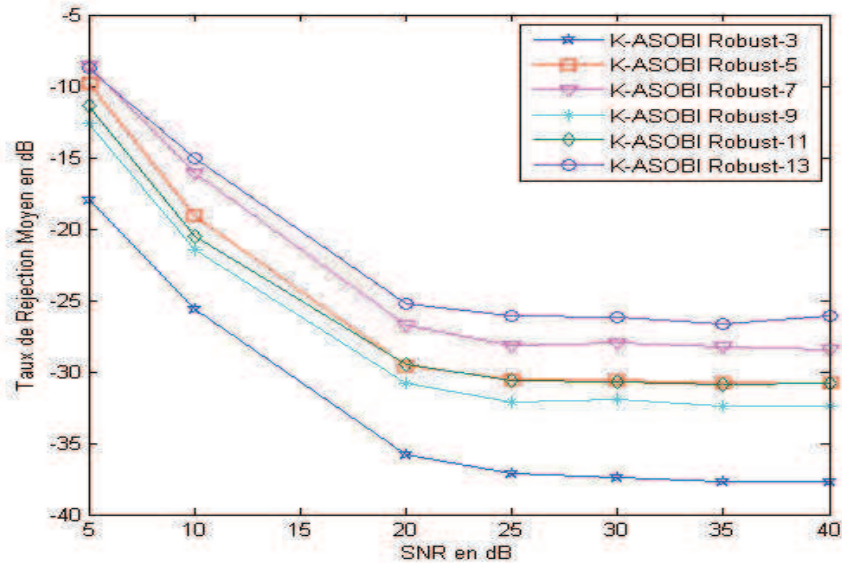


Figure 3.5 — Taux de réjection moyen vs SNR. Robust k-ASOBI

3.4.2 Performance de l'algorithme Robust k-ASOBI

Des simulations similaires sont effectuées dans les mêmes conditions que la section 3.4.1, en traçant des courbes de performances de l'algorithme k-ASOBI pour plusieurs valeurs de k . En effet, sur la figure 3.5, le taux de réjection I_{perf} est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille 1024.

Dans la figure 3.6, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l'échantillon. Les figures 3.5 et 3.6 indiquent une amélioration significative des performances pour un nombre suffisamment grand d'échantillons et un SNR élevé. On remarque que quand les valeurs de k décroissent (de 9 à 3 dans nos simulations) les performances s'améliorent, ceci s'explique par le fait que plus le retard k est proche de 0 ; plus le signal possède de l'énergie.

3.4.3 Performance de l'algorithme Robust Optimal k-ASOBI

Des simulations similaires sont effectuées dans les mêmes conditions que la section 3.4.2, en traçant des courbes de performances de l'algorithme Optimal k-ASOBI. En effet, sur la figure 3.7, le taux de réjection I_{perf} est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille 1024 .

Dans la figure 3.8, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l'échantillon. Les figures 3.7 et 3.8 indiquent une amélioration significative des performances pour un nombre suffisamment élevé d'échantillons et un SNR élevé. Ainsi on remarque que les trois courbes se confondent car, à chaque fois, l'algorithme Optimal k-ASOBI calcule la valeur de k qui permet d'avoir les meilleures performances.

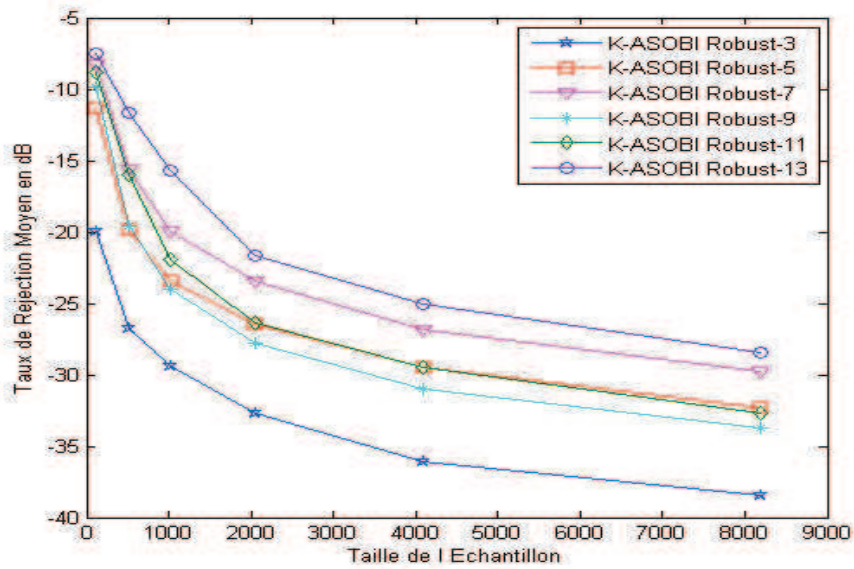


Figure 3.6 — Taux de réjection moyen vs Taille de l'échantillon. Robust k-ASOBI

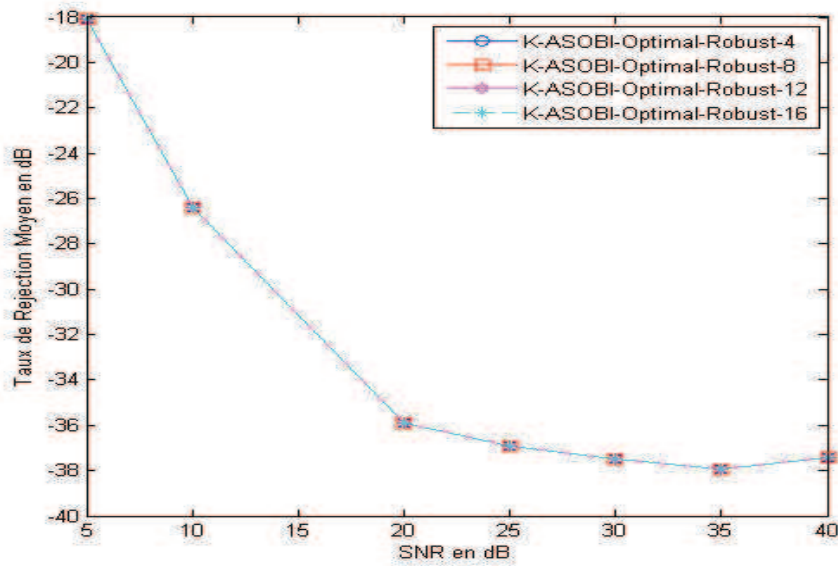


Figure 3.7 — Taux de réjection moyen vs SNR.

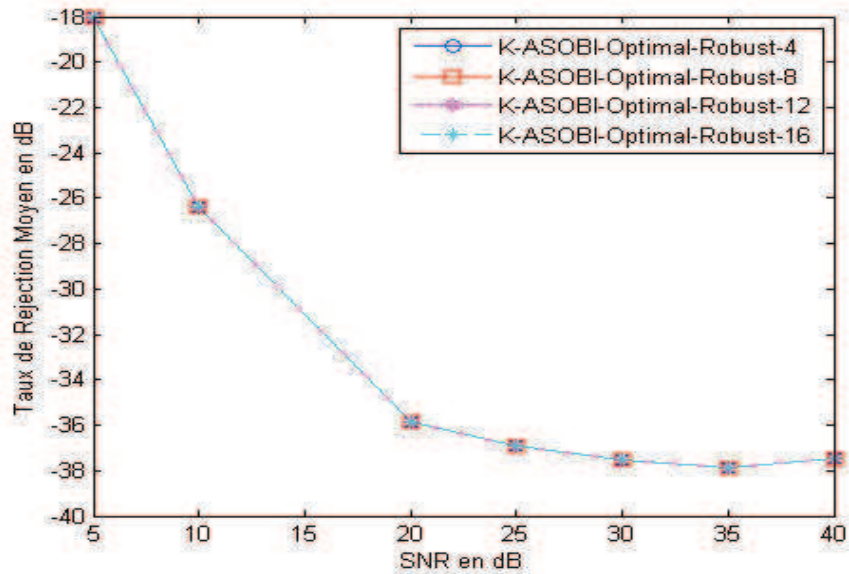


Figure 3.8 — Taux de réjection moyen vs Taille de l'échantillon.

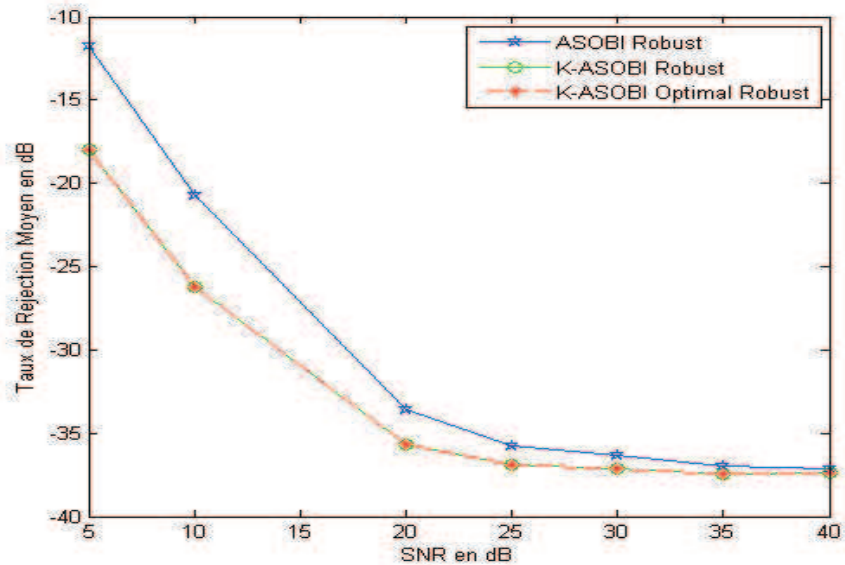


Figure 3.9 — Taux de réjection moyen vs SNR.

3.4.4 Comparaison entre les différentes versions de Robust ASOBI

Dans cette section, une comparaison est effectuée entre les trois algorithmes (Robust ASOBI, Robust k-ASOBI et Robust k-ASOBI Optimal) pour $k=4$, les simulations effectuées dans les mêmes conditions que les sections précédentes. En effet, les figures 3.9 et 3.10 montrent que les algorithmes Robust K-ASOBI et Robust K-ASOBI optimal se confondent et présentent les meilleures performances car $k=4$ est la valeur optimale.

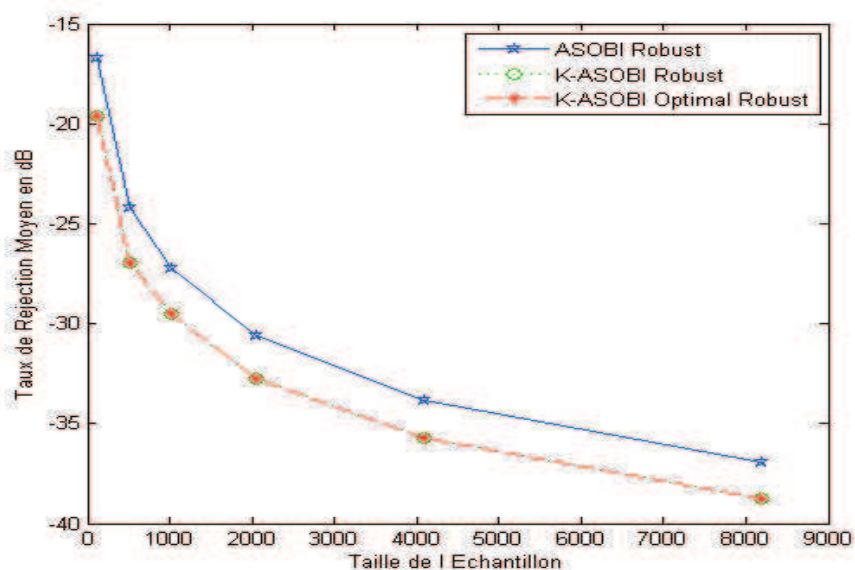


Figure 3.10 — Taux de réjection moyen vs Taille de l'échantillon.

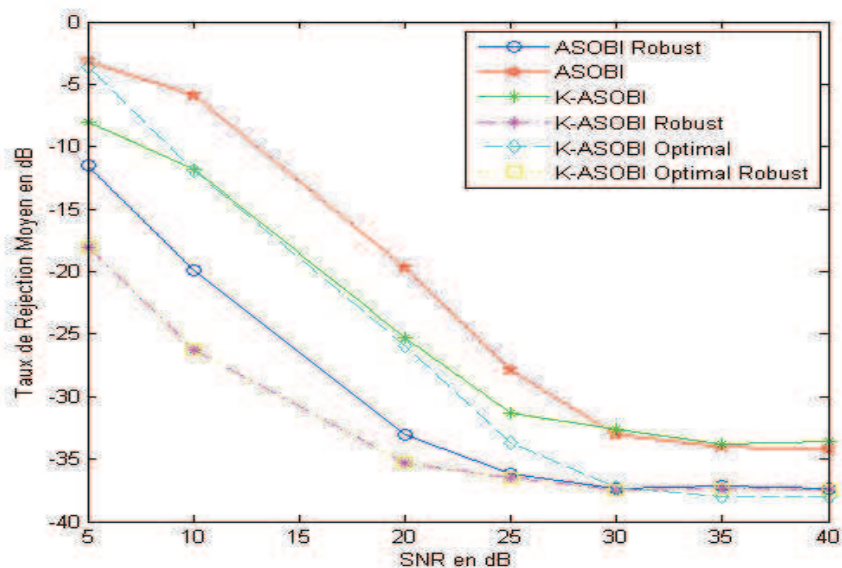


Figure 3.11 — Taux de réjection moyen vs SNR.

3.4.5 Comparaison entre les différentes versions de ASOBI et Robust ASOBI

Dans cette section, une comparaison est effectuée entre les six algorithmes (ASOBI, k-ASOBI, Optimal k-ASOBI, Robust ASOBI, Robust k-ASOBI et Robust k-ASOBI Optimal) pour $k=4$. En effet, les figures 3.11 et 3.12 montrent que les versions de l'algorithme Robust ASOBI donnent les meilleures performances par rapport aux versions de l'algorithme ASOBI.

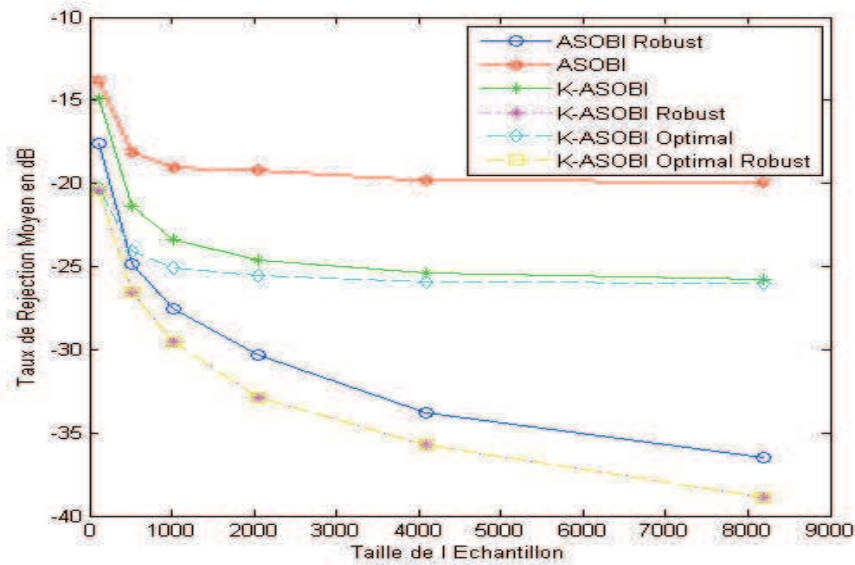


Figure 3.12 — Taux de réjection moyen vs Taille de l'échantillon.

3.5 Conclusion

Les algorithmes développés au chapitre 2 nécessitent pour la connaissance ou l'estimation de la variance du bruit, l'estimation des corrélations des signaux reçus. Pour éviter l'estimation de la variance du bruit, nous avons proposé dans ce chapitre une deuxième classe d'algorithmes qui n'exploitent que les corrélations des signaux à des retards différents de zéro sous l'hypothèse d'un bruit temporellement blanc. Ces algorithmes ainsi développés dans ce chapitre sont alors aussi exploitables dans le cas d'un bruit spatialement corrélé pourvu qu'il soit temporellement blanc. Les algorithmes proposés dans ce chapitre gardent les mêmes avantages de simplicité que ceux présentés au chapitre précédent. Ils sont ainsi facilement implémentables sur les processeurs du traitement du signal. Les simulations ont montré que les algorithmes développés présentent de bonnes performances.

4 Implémentation adaptative

Dans ce chapitre, une version adaptative simple de l'algorithme ASOBI, dans un contexte variable dans le temps, est proposée. A cet effet, nous proposons un algorithme adaptatif spécifique pour l'identification aveugle d'un système à deux entrées et deux sorties. A partir d'une solution analytique de l'identification aveugle du système exploitant les propriétés de cohérence temporelle des sources, nous présentons une mise en œuvre adaptative simplifiée de l'algorithme ASOBI (Analytical Second Order Blind Identification) pour le calcul d'un séparateur à faible complexité. Deux versions adaptatives sont alors proposées. De même une version adaptative de l'algorithme k-ASOBI est proposée. Des simulations numériques sont présentées pour l'évaluation des performances des algorithmes adaptatifs proposés.

4.1 L'algorithme ASOBI Adaptatif

Reprenons le modèle :

$$\mathbf{x}(t) = \mathbf{H} \mathbf{s}(t) + \mathbf{n}(t), \quad (4.1)$$

Où $\mathbf{s}(t)$ est un vecteur 2×1 , la i^{eme} entrée notée $s_i(t)$ correspond au signal émis par la i^{me} source. La matrice 2×2 du canal inconnu est donnée par

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix}$$

\mathbf{H} est supposée être de rang plein. Les signaux sources sont temporellement colorés et sont des processus mutuellement non corrélés. En outre, nous supposons que le bruit $\mathbf{n}(t)$ n'est pas corrélé avec les signaux source et temporellement blanc avec une matrice de covariance $\mathbb{E}[\mathbf{n}(t)\mathbf{n}(t)^T] = \sigma^2\mathbf{I}$, où $\mathbb{E}[\cdot]$ dénote l'espérance mathématique et \mathbf{I} la matrice identité 2×2 .

Pour rappel, dans la version par blocs de l'algorithme ASOBI, les corrélations sont estimées par :

$$\hat{r}_{ij}(k) = \frac{1}{N} \sum_{n=k}^{N-1} x_i(n)x_j(n-k), \quad i, j = 1, 2. \quad (4.2)$$

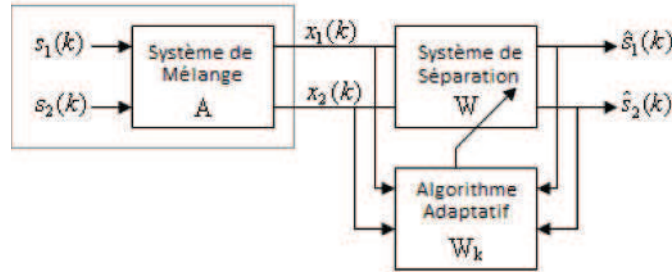


Figure 4.1 — Schéma illustrant l'algorithme ASOBI adaptatif

Pour obtenir une version adaptative de l'algorithme ASOBI, les corrélations sont estimées en utilisant un estimateur bloc par bloc [40]. La figure 4.1 montre le schéma de l'algorithme proposé.

Écrivons les données observées comme suit :

$$\mathbf{x}_i = \left(\underbrace{x_i(0), \dots, x_i(M-1)}_{\text{block 1}} \mid \underbrace{x_i(M), \dots, x_i(2M-1)}_{\text{block 2}} \mid \dots \mid \underbrace{x_i(N-M), \dots, x_i(N-1)}_{\text{block } l} \right), \quad i = 1, 2. \quad (4.3)$$

Où M est la taille du bloc et l est le numéro du bloc.

Les corrélations sont estimées de façon adaptative en utilisant l'expression suivante :

$$\hat{r}_{ij}^{(l+1)}(k) = \psi \hat{r}_{ij}^{(l)}(k) + \frac{1-\psi}{M} \sum_{n=lM+1+k}^{M(l+1)} x_i(n)x_j(n-k), \quad i, j = 1, 2. \quad (4.4)$$

Où ψ est le facteur d'oubli avec $0 \leq \psi \leq 1$.

Une fois les corrélations estimées à partir de l'équation (4.2), on calcule \mathbf{F}_{ij} , \mathbf{T}_{ij} , \mathbf{F}_i et \mathbf{T}_j pour chaque nouveau bloc l de la même manière que celle décrite dans les sections précédentes. Une estimation de la matrice de mélange est calculée de manière récursive comme suit :

$$\hat{\mathbf{H}}(l+1) = \phi \hat{\mathbf{H}}(l) + (1-\phi) \hat{\mathbf{H}}_s \quad (4.5)$$

Où $\hat{\mathbf{H}}_s$ est la version échantillonnée de \mathbf{H}_s , donnée par l'équation :

$$\hat{\mathbf{H}}_s = \begin{pmatrix} 2a.\mathbf{F}_1 - \mathbf{T}_1(b-c) & 2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b+c) \\ 2a.\mathbf{F}_{12} - \mathbf{T}_{12}(b-c) & 2a.\mathbf{F}_2 - \mathbf{T}_2(b+c) \end{pmatrix}; \quad (4.6)$$

Sachant que ϕ est le facteur d'oubli avec $0 \leq \phi \leq 1$. Une estimation de la matrice de séparation est aussi calculée de manière récursive :

$$\hat{\mathbf{W}}(l+1) = \phi \hat{\mathbf{W}}(l) + (1-\phi) \hat{\mathbf{W}}_s \quad (4.7)$$

Où $\hat{\mathbf{W}}_s$ est la version estimée de \mathbf{W} déduite de $\hat{\mathbf{H}}_s$ selon l'équation suivante

$$\hat{\mathbf{W}}_s = \begin{pmatrix} h_{22} & -h_{12} \\ -h_{21} & h_{11} \end{pmatrix}; \quad (4.8)$$

L'algorithme, appelé ASOBIadap1, est résumé dans le tableau Algorithm 7.

Algorithm 7 Résumé de l'algorithme adaptatif Asobi (ASOBIadap1)

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$ pour le bloc l avec une longueur M ;
 - 2: **Estimer** : $\hat{r}_{ij}^{(l)}(k)$, $i, j = 1, 2$ avec $k = 0, \dots, K$ pour l blocs en utilisant l'équation (4.4) ;
 - 3: **Calculer** : $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{12}, \mathbf{F}_1, \mathbf{F}_2$ et \mathbf{F}_{12} pour l blocs en utilisant les équations de (2.10) à (2.15) ;
 - 4: **Calculer** : b, c, d en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Calculer** : la matrice $\hat{\mathbf{W}}_s$ et $\hat{\mathbf{H}}_s$ pour l blocs selon l'équation selon les équations (2.50) et (2.51) ;
 - 6: **Mettre à jour** : $\hat{\mathbf{W}}(l)$ et $\hat{\mathbf{H}}(l)$ selon les équations (4.5) et (4.7) ;
 - 7: **Retour** : à l'étape 1 pour les nouveaux signaux observés $\mathbf{x}_1, \mathbf{x}_2$ du bloc suivant.
-

4.1.1 Traitement adaptatif efficace

Notons que l'algorithme adaptatif présenté ci-dessus ne fonctionne pas aussi bien que le ASOBI classique (voir les figures 4.4 et 4.5). Pour pallier cette inefficacité, nous proposons d'utiliser une version modifiée d'estimateur dans l'équation (4.4). Pour cet objectif, nous utilisons des blocs qui se chevauchent avec $(M - 1)$ échantillons [41]. Autrement dit, au lieu d'observer M nouveaux échantillons, nous observons uniquement un seul nouvel échantillon et on le groupe avec les derniers $(M - 1)$ échantillons observés pour former notre bloc. Ceci est illustré dans l'équation suivante :

$$\hat{r}_{ij}^{(l+1)}(k) = \psi \hat{r}_{ij}^{(l)}(k) + \frac{1 - \psi}{M} \sum_{i=l+1+k}^{l+M} x_i(n)x_j(n - k), \quad i, j = 1, 2. \quad (4.9)$$

L'algorithme amélioré, dénommé ASOBIadap2, est résumé dans le tableau Algorithm 8.

Algorithm 8 Résumé de l'algorithme adaptatif amélioré (ASOBIadap2)

- 1: **Observer** les vecteurs $\mathbf{x}_1, \mathbf{x}_2$ pour l blocs avec longueur M ;
 - 2: **Estimer** $\hat{r}_{ij}^{(l)}(k)$, $i, j = 1, 2$ avec $k = 0, \dots, K$ pour l blocs en utilisant l'équation (4.9) ;
 - 3: **Calculer** : $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{12}, \mathbf{F}_1, \mathbf{F}_2$ et \mathbf{F}_{12} pour l blocs en utilisant les équations de (2.10) à (2.15) ;
 - 4: **Calculer** : b, c, d en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Calculer** : la matrice $\hat{\mathbf{W}}_s$ et $\hat{\mathbf{H}}_s$ pour l blocs selon les équations (2.50) et (2.51) ;
 - 6: **Retour** : à l'étape 1 pour les nouveaux signaux observés $\mathbf{x}_1, \mathbf{x}_2$ du bloc suivant.
-

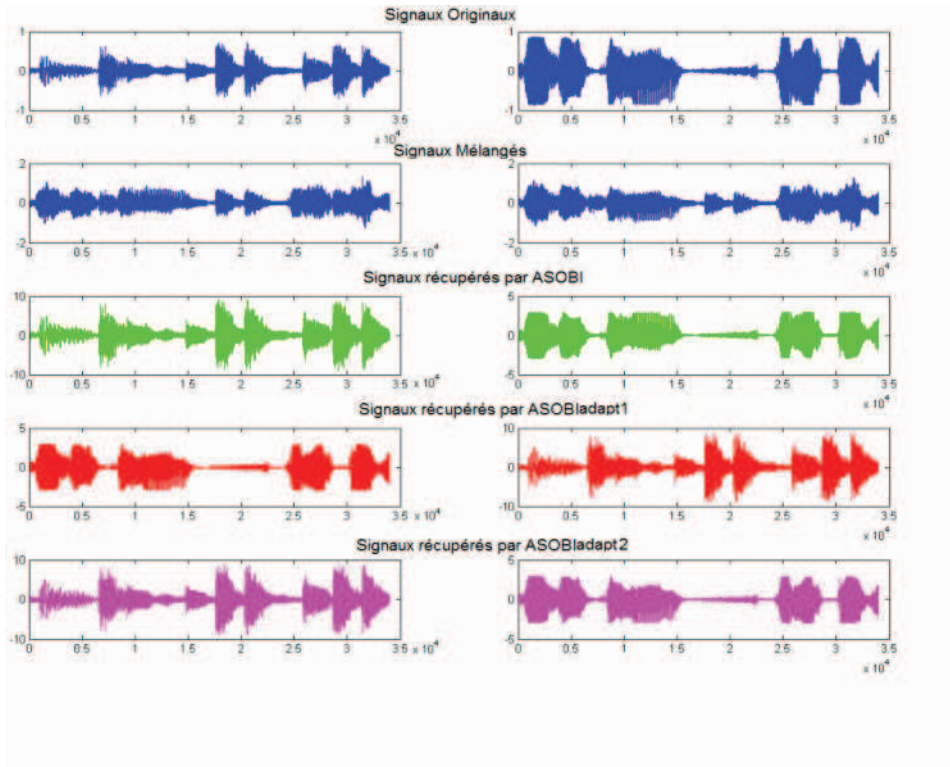


Figure 4.2 — Résultat de la simulation : signaux de parole; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l’Algorithme Asobi, Ligne 4 : Signaux Récupérés par l’Algorithme ASOBIadapt1 , Ligne 5 : Signaux Récupérés par l’Algorithme ASOBIadapt2

4.1.2 Résultats de simulation

Tout d’abord, une série d’analyses des solutions proposées est présentée : Deux signaux de parole échantillonnés à 8000 Hz sont mélangés par une matrice aléatoire \mathbf{H} .

Les tracés des deux signaux individuels de parole et leurs mélanges sont représentés sur la figure 4.2. L’algorithme ASOBI [27] est utilisé pour la comparaison, de même pour la figure 4.3 où deux signaux EEG et ECG sont utilisés. Il est clair que les solutions adaptatives proposées fonctionnent bien dans ce cas et donnent des résultats similaires à l’algorithme ASOBI.

4.1.3 Évaluation des performances

Considérons deux sources mélangées par la matrice aléatoire \mathbf{H} . Le bruit additif est un processus gaussien de moyenne zéro et de matrice de covariance $\sigma^2 \mathbf{I}_N$. Les signaux sources ont une variance unité et chacun est généré par le filtrage d’un processus gaussien blanc à l’aide de deux filtres autorégressifs différents. Le taux de réjection global est moyenné sur 100 réalisations. Dans ces simulations, les solutions proposées supposent que le bruit est absent (c’est-à-dire σ^2 est mis à zéro) et $P = 8$. La convergence de la matrice de corrélation se produit après chaque bloc de longueur $M = 50$ échantillons. Sur la figure 4.4, le taux de réjection I_{perf} (pour ASOBI, ASOBIadapt1 et ASOBIadapt2)

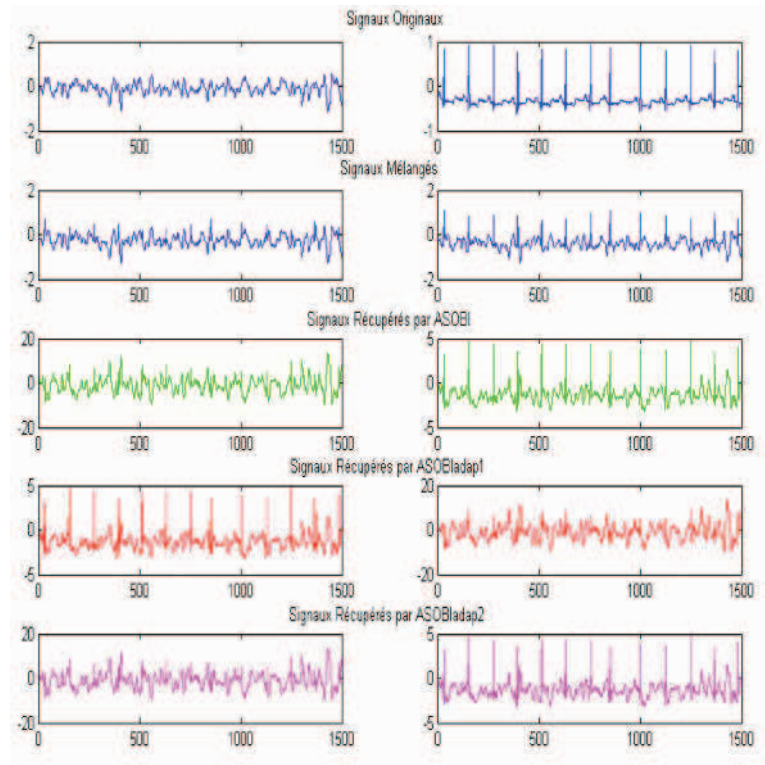


Figure 4.3 — Résultat de la simulation : signaux EEG et ECG ; Ligne 1 : Signaux Originaux, Ligne 2 : Signaux Mélangés, Ligne 3 : Signaux Récupérés par l'Algorithme Asobi, Ligne 4 : Signaux Récupérés par l'Algorithme ASOBIadap1 , Ligne 5 : Signaux Récupérés par l'Algorithme ASOBIadap2

est tracé en dB en fonction du rapport signal sur bruit (SNR) en dB pour un échantillon de taille de 1024 .

Dans la figure 4.5, le SNR est maintenu à 20 dB . Les courbes montrent le taux de réjection I_{perf} en dB en fonction de la taille de l'échantillon. Les figures 4.4 et 4.5 indiquent une amélioration significative des performances pour un nombre suffisamment grand d'échantillons et un SNR élevé avec les facteurs $\psi = 0.8$ et $\phi = 0.85$. En outre, on peut observer que la solution ASOBI par bloc et sa version adaptative proposée ont des performances similaires. Les trois courbes des figures 4.4 et 4.5 ont presque la même forme, mais ASOBIadap2 s'aligne parfaitement avec ASOBI. Lorsque l'on compare la version adaptative ASOBIadap1 par rapport à l'algorithme ASOBI par bloc, on observe une perte de performance d'environ 4 dB pour un $SNR = 20$ et $N = 1024$ échantillons mais avec une réduction importante du coût de calcul. Ce dernier est réduit par un facteur correspondant à la taille du bloc qui est égal à M .

La figure 4.6 présente le taux de réjection I_{perf} en fonction de la taille de l'échantillon de l'algorithme adaptatif pour différentes longueurs de blocs M . Nous pouvons clairement observer une amélioration significative des performances pour des longueurs de bloc élevées.

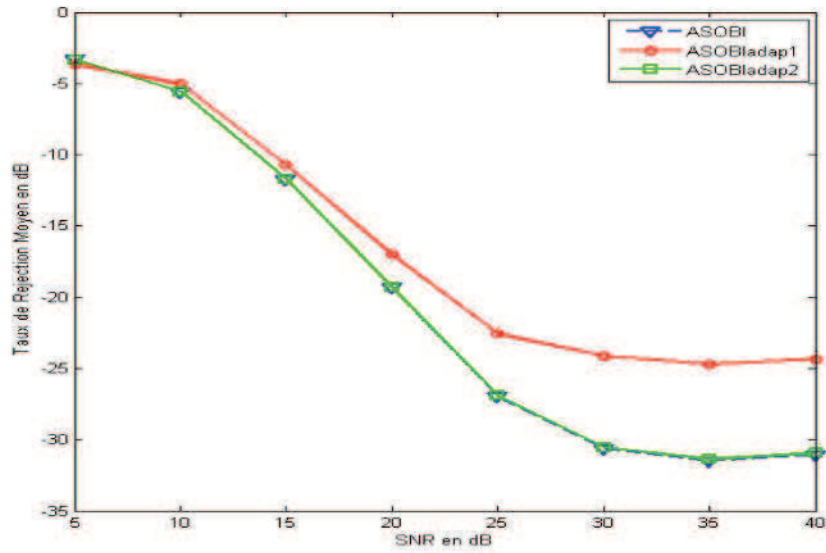


Figure 4.4 — Taux de réjection moyen vs SNR pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$.

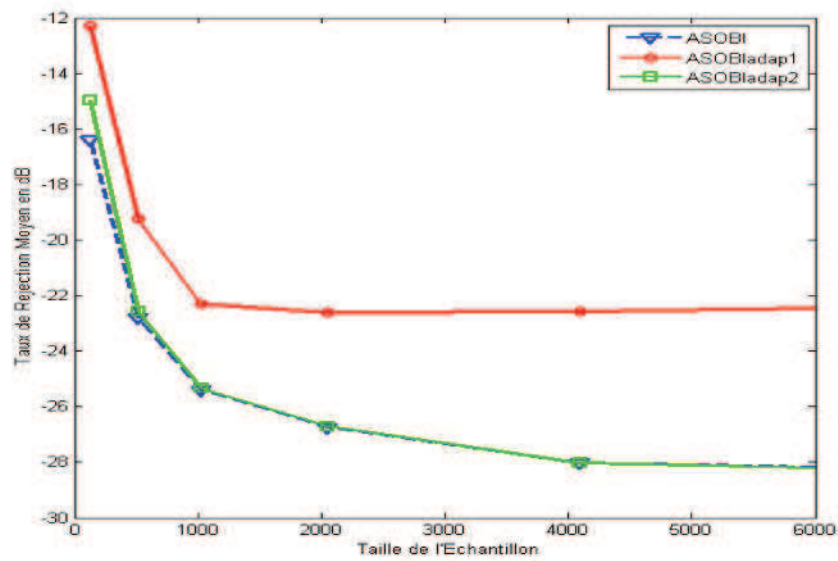


Figure 4.5 — Taux de réjection moyen vs Taille de l'échantillon pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$.

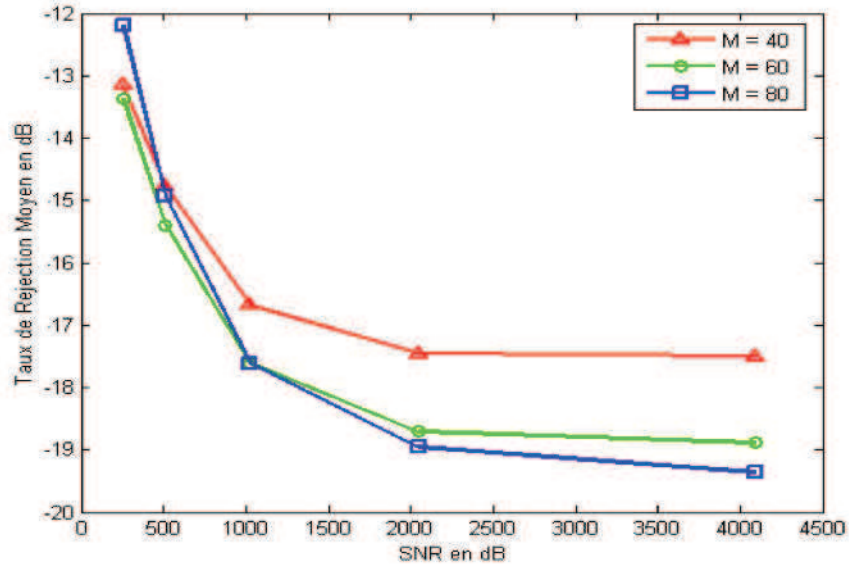


Figure 4.6 — Taux de réjection moyen vs Taille de l'échantillon pour plusieurs longueur du bloc (cas de ASOBIadap2).

4.1.4 Complexité calculatoire

Dans cette section la complexité de calcul des trois algorithmes SAS proposés ci-dessus est donnée. Pour le traitement de N échantillons, cette complexité en terme de multiplication est calculée comme suit :

$$C_{ASOBI} = 3((P + 1)N - P(P + 1)/2) + 21 \quad (4.10)$$

$$C_{ASOBIadap1} = \frac{N}{M} [3((P + 1)M - P(P + 1)/2) + 21] \quad (4.11)$$

$$C_{ASOBIadap2} = [3(P + 1) + 21](N - M) + (3((P + 1)M - P(P + 1)/2) + 21) \quad (4.12)$$

Où M : Longueur du bloc et P : nombre de temps de retards considérés.

A noter que la complexité ci-dessus est donnée pour le traitement du nombre total de N échantillons. Dans le présent processus, les versions adaptatives nécessitent beaucoup moins de ressources en terme de multiplications que la version de traitement par blocs, à savoir $3((P + 1)M - P(P + 1)/2) + 21$ et $3(P + 1) + 21$ pour le ASOBIadap1 et ASOBIadap2, respectivement.

Ensuite, nous comparons la complexité de l'algorithme ASOBI avec les deux versions adaptatives de ASOBI (ASOBIadap1 et ASOBIadap2). Dans la figure 4.7, la complexité est calculée en terme du nombre de multiplications en fonction de la taille de l'échantillon pour une longueur de bloc de 50 échantillons et $P = 8$. Le nombre d'opérations augmente plus rapidement lors de l'utilisation du ASOBIadap2, tandis que ASOBIadap1 a un peu moins de calculs que le ASOBI par blocs.

Dans la figure 4.8, la complexité est tracée en fonction de la longueur du bloc, la

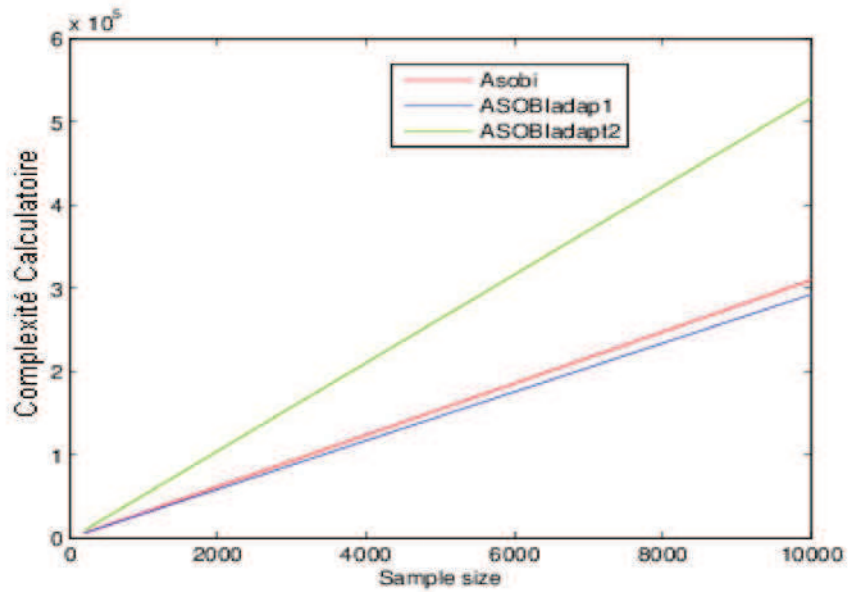


Figure 4.7 — Complexité de calcul vs taille de l'échantillon.

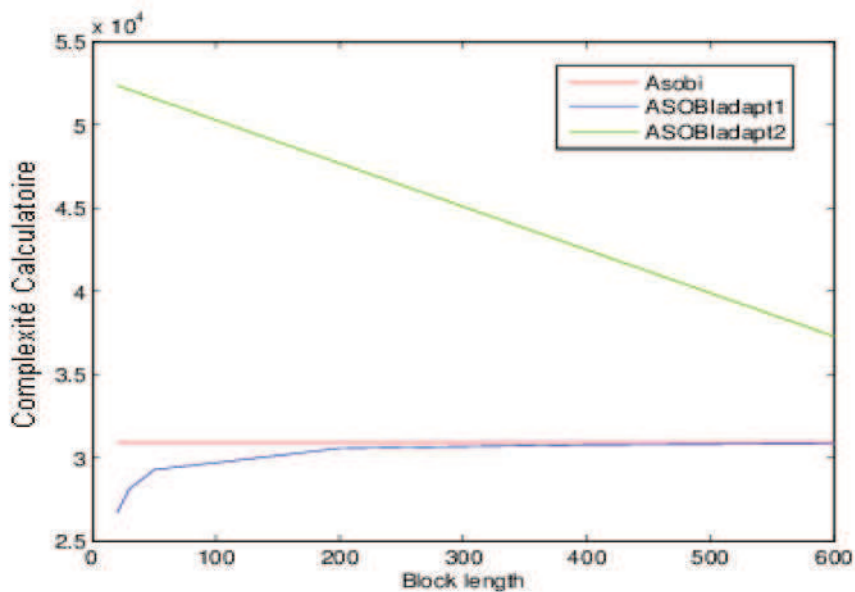


Figure 4.8 — Complexité de calcul vs longueur du bloc.

taille de l'échantillon est fixée à 1000. Il apparaît que la complexité de ASObIadap1 augmente avec la longueur du bloc et se rapproche de ASObI par blocs, tandis que la complexité de l'ASObIadap2 diminue avec la longueur du bloc. Nous observons également que ASObIadap2 a besoin de plus de calcul que ASObIadap1. Comme ASObIadap2 fonctionne mieux que ASObIadap1, un compromis doit être fait entre la performance et la complexité numérique.

4.2 La version adaptative de l'algorithme K-ASOBI

Dans le prolongement de la section 4.1, il est proposé dans ce qui suit de procéder de la même façon pour obtenir les deux versions adaptatives de l'algorithme k-ASOBI :

L'algorithme, appelé k-ASOBIadap1, est résumé dans le tableau Algorithm 9.

Algorithm 9 Résumé de l'algorithme adaptatif k-ASOBI (k-ASOBIadap1)

- 1: **Observer** : les vecteurs $\mathbf{x}_1, \mathbf{x}_2$ pour le bloc l avec une longueur M ;
 - 2: **Estimer** : $\hat{r}_{ij}^{(l)}(k)$, $i, j = 1, 2$ avec $k = 0, \dots, K$ pour un choix arbitraire du retard k et en utilisant l blocs en utilisant l'équation (4.4) ;
 - 3: **Calculer** : $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{12}, \mathbf{F}_1, \mathbf{F}_2$ et \mathbf{F}_{12} pour l blocs en utilisant les équations de (2.52) à (2.57) ;
 - 4: **Calculer** : b, c, d en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Calculer** : la matrice $\hat{\mathbf{W}}_s$ et $\hat{\mathbf{H}}_s$ pour l blocs selon l'équation selon équation (2.60) ;
 - 6: **Mettre à jour** : $\hat{\mathbf{W}}(l)$ et $\hat{\mathbf{H}}(l)$ selon les équations (4.5) et (4.7) ;
 - 7: **Retour** : à l'étape 1 pour les nouveaux signaux observés $\mathbf{x}_1, \mathbf{x}_2$ du bloc suivant.
-

L'algorithme amélioré, dénommé k-ASOBIadap2, est résumé dans le tableau Algorithm 10.

Algorithm 10 Résumé de l'algorithme adaptatif amélioré (k-ASOBIadap2)

- 1: **Observer** les vecteurs $\mathbf{x}_1, \mathbf{x}_2$ pour un choix arbitraire du retard k et en utilisant l blocs avec longueur M ;
 - 2: **Estimer** $\hat{r}_{ij}^{(l)}(k)$, $i, j = 1, 2$ avec $k = 0, \dots, K$ pour l blocs en utilisant l'équation (4.9) ;
 - 3: **Calculer** : $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{12}, \mathbf{F}_1, \mathbf{F}_2$ et \mathbf{F}_{12} pour l blocs en utilisant les équations de (2.52) à (2.57) ;
 - 4: **Calculer** : b, c, d en utilisant les équations de (2.38) à (2.40) ;
 - 5: **Calculer** : la matrice $\hat{\mathbf{W}}_s$ et $\hat{\mathbf{H}}_s$ pour l blocs selon l'équation selon équation (2.60) ;
 - 6: **Retour** : à l'étape 1 pour les nouveaux signaux observés $\mathbf{x}_1, \mathbf{x}_2$ du bloc suivant.
-

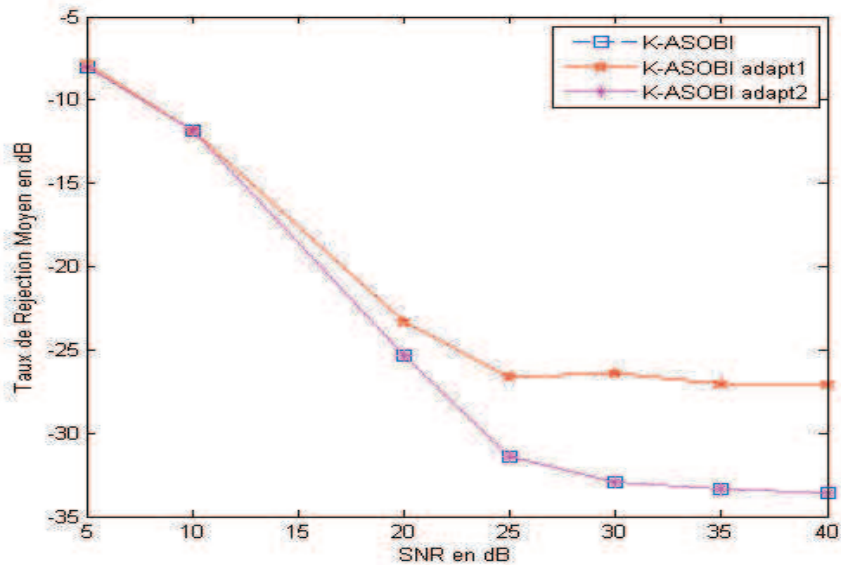


Figure 4.9 — Taux de réjection moyen vs SNR pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$.

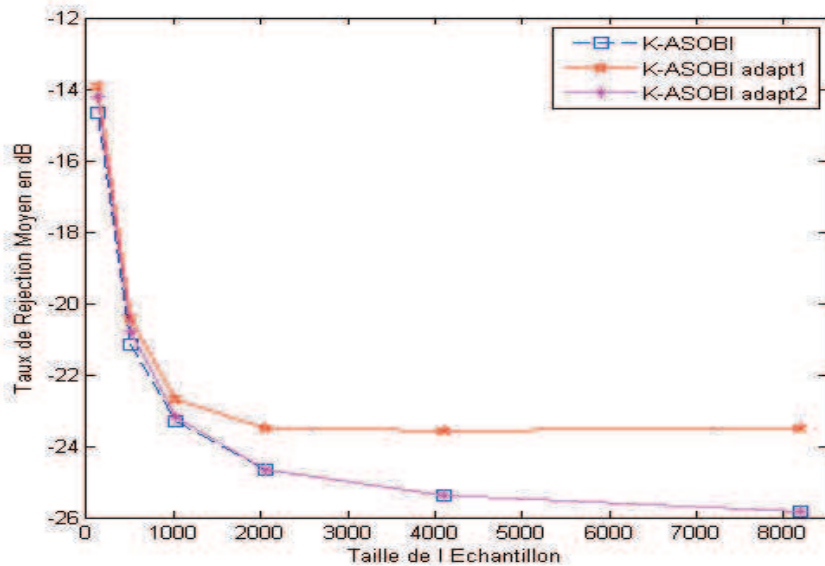


Figure 4.10 — Taux de réjection moyen vs Taille de l'échantillon pour le facteur de mémoire $\psi = 0.8$, $\phi = 0.85$.

4.3 Conclusion

Dans ce chapitre nous avons proposé une solution adaptative efficace à faible complexité calculatoire destinée pour le problème spécifique de la séparation de deux signaux à partir de deux mélanges instantanés. La solution proposée tend à résoudre le problème de la séparation aveugle à travers les statistiques du second ordre en sup-

4.3. CONCLUSION

posant que les signaux sources soient temporellement corrélés. Elle exploite l'avantage de l'indétermination inhérente au traitement aveugle pour aboutir à des algorithmes sans opérations de division. Ces algorithmes sont appropriés pour implémentation sur les DSP (Digital Signal Processor) et les FPGA (Field programmable Gate Array) où une division nécessite des ressources importantes. À travers les simulations réalisées, il est montré que la solution adaptative des algorithmes proposées (ASOBIadapt2, k-ASOBIadapt2) présentent respectivement des performances similaires avec un ajustement parfait pour l'algorithme ASOBI et k-ASOBI.

Implémentation temps réel sur carte DSK C6713

Les performances d'un algorithme dépendent de la taille du programme d'une part, et du cycle du processeur utilisé d'autre part. Le temps d'exécution d'un programme donne une idée sur l'efficacité des algorithmes et la possibilité d'implémentation en temps réel [42]. Dans cette thèse, la partie implémentation consiste à programmer deux algorithmes en temps réel par un processeur de traitement du signal, en l'occurrence le TMS320C6713.

5.1 Notions sur les DSP

Le DSP (Digital Signal Processor) fait partie d'un type particulier de microprocesseurs. Ses fonctions spéciales ainsi que son architecture le rendent performant dans le domaine du traitement du signal. Ces caractéristiques ont augmenté le champ d'applications (communications, imagerie, applications militaires). Il existe plusieurs fabricants des DSP comme Analog Devices, Motorola et le leader du marché Texas Instruments [43].

Texas Instruments a introduit la première génération TMS320C10 digital signal processor en 1982, le TMS320C25 en 1986, et le TMS320C50 en 1991. Plusieurs versions de chacun de ces processeurs C1x, C2x et C5x existent avec différentes caractéristiques, comme la vitesse d'exécution. Ces processeurs à 16 bits sont tous des processeurs à virgule fixe, et ont un code compatible. Le processeur TMS320C30 à virgule flottante est introduit à la fin des années quatre-vingt. Le C31, C32 et le plus récent C33 font parties de la famille C3x des processeurs à virgule flottante. Les C4x le sont aussi, introduits ultérieurement, possèdent un code compatible avec les processeurs C3x qui sont tous basés sur l'architecture de Harvard modifiée [44]. Il est important de faire rappeler l'avantage d'une telle architecture, qui donne un bon compromis entre l'architecture de VonNeumann pénalisante par ses faibles performances en matière de cycles d'exécution des instructions, et l'architecture de Harvard en raison de son prix élevé. Le TMS320C6201 (C62x), présenté en 1997, est le premier membre de la famille C6x des DSP à virgule fixe. Contrairement à ses prédécesseurs (C1x, C2x et C5x), le C62x est basé sur l'architecture very-long-instruction-word (VLIW) qui prend toujours en considération l'architecture de Harvard modifiée. L'architecture VLIW dispose d'un jeu d'instructions simple. Le code du C62x n'est pas compatible avec les générations

précédentes des processeurs virgule fixe. Ultérieurement, le processeur à virgule flottante TMS320C6701 (C67x) est introduit comme un autre membre de la famille C6x des DSP TI. Le jeu d'instructions du C62x est la base du jeu d'instructions C67x. Une addition récente aux processeurs de la famille C6x est le DSP virgule fixe C64x. Le DSP virgule fixe est privilégié dans le cas d'applications (ou périphériques) utilisant des batteries, comme les téléphones cellulaires, car il consomme moins d'énergie qu'un équivalent en virgule flottante. Contrairement aux processeurs 16 bits à virgule fixe C1x, C2x et C5x, le DSP C6x virgule fixe (32 bits) présente une amélioration de la dynamique et de la précision. Il est à noter qu'avec ce type de processeur, il faut mettre les données à l'échelle et prendre en considération le débordement. Le processeur virgule flottante est en général très coûteux par son nombre de broches, le nombre de bus qu'il intègre ainsi que les circuits additionnels nécessaires pour tenir compte de l'arithmétique en virgule fixe. Plusieurs facteurs comme le prix, la puissance consommée et la vitesse de traitement interviennent dans le choix du DSP [34].

5.2 Plate-forme de développement

5.2.1 L'interface graphique Code Composer Studio

L'outil logiciel commercialisé par Texas Instruments réunissant les outils d'édition, d'optimisation, de compilation, d'assemblage et d'éditeurs de liens s'appelle Code Composer Studio. Cet outil se manipule par l'intermédiaire d'une interface graphique multifenêtre configurable, compatible Windows et commune à d'autres familles de DSP de Texas Instruments (Figure 5.1). Installé sur un ordinateur de type PC, il permet aussi de déboguer les programmes et de simuler le DSP sur le calculateur hôte. Code Composer Studio possède de manière non exhaustive les fonctionnalités suivantes :

- paramétrage des niveaux d'optimisation du C avec visualisation du code assembleur généré ;
- visualisation et modification en cours d'exécution du contenu de la mémoire interne et des registres du DSP ainsi que des variables des programmes en C et en assembleur ;
- affichage en cours d'exécution des résultats des traitements par l'intermédiaire de fenêtres graphiques paramétrables en représentation temporelle, fréquentielle (FFT), plan complexe (constellation) ;
- simulation d'interruption (points de sonde) avec transfert de données en entrée/sortie entre la mémoire interne du DSP et des fichiers notamment au format texte codé selon la norme ASCII ;
- points d'arrêts, exécution pas à pas avec sauts paramétrables ;
- évaluation des temps d'exécution de parties sélectionnées (points d'horloge) sur le programme C et assembleur avec affichage en secondes ou en temps cycles.

Nous allons nous intéresser maintenant au Kit de développement DSK C6211 muni du processeur DSP TMS320 C6211 (voir figure 5.2 et 5.3).

5.2. PLATE-FORME DE DÉVELOPPEMENT

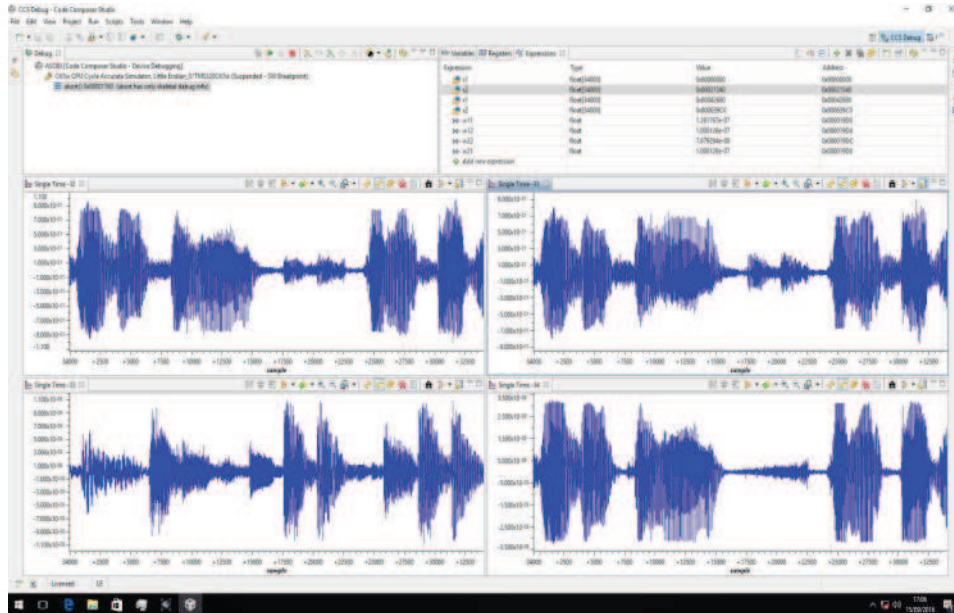


Figure 5.1 — Interface graphique Code Composer Studio.

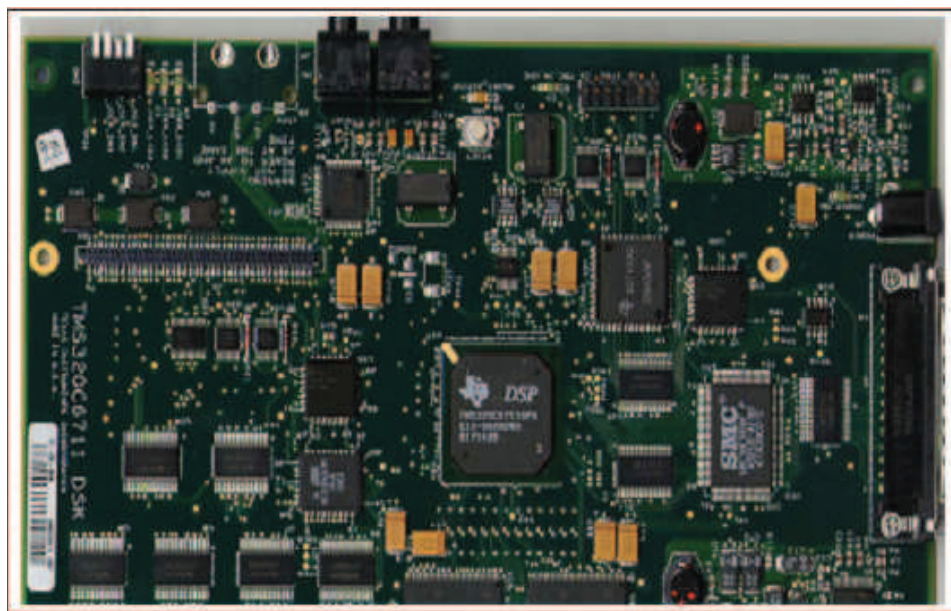


Figure 5.2 — La carte DSK C6713.

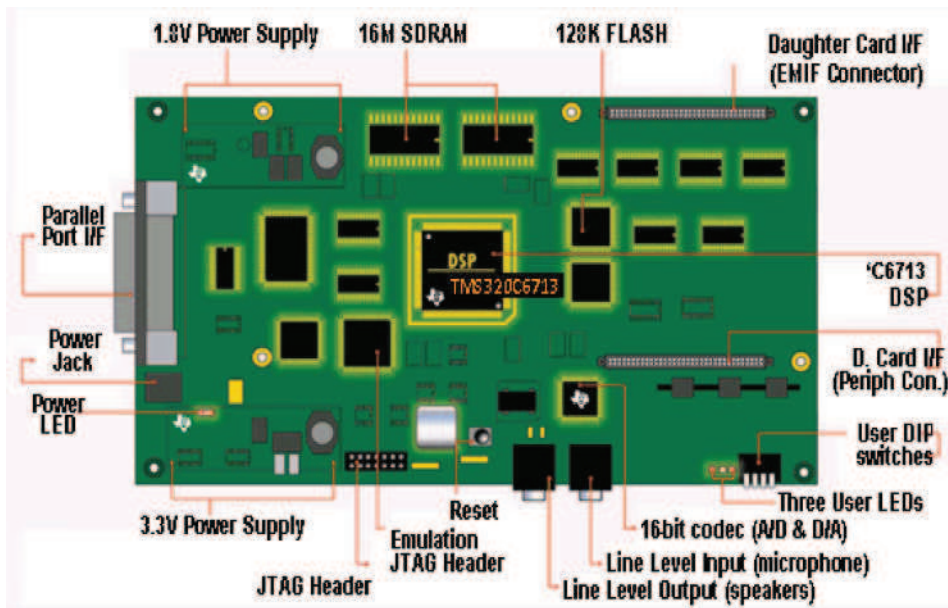


Figure 5.3 — Détail de la carte DSK C6713.

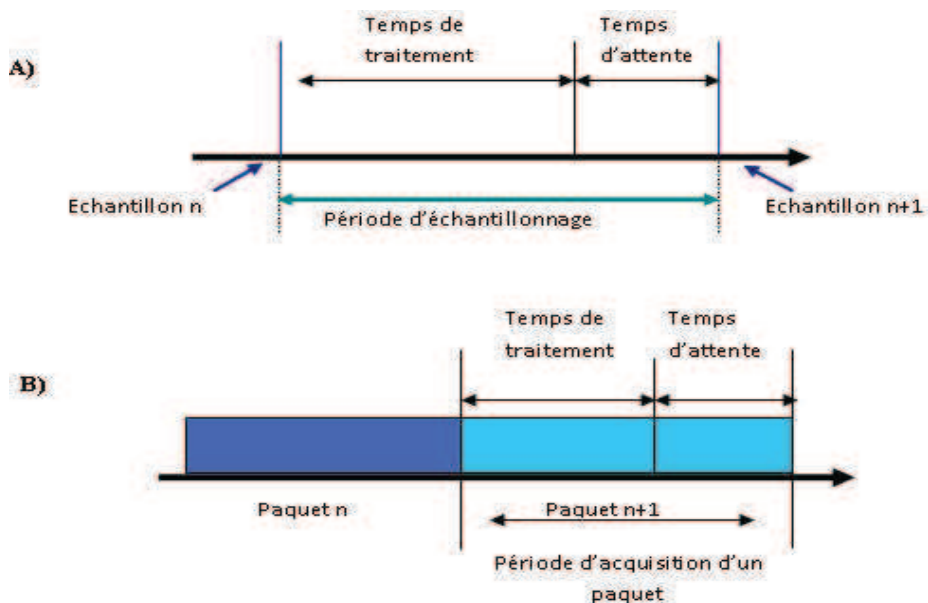


Figure 5.4 — Notion du temps réel. A) Traitement par échantillon. B) Traitement par paquet.

5.2.2 Définition du temps réel

La définition du temps réel dépend de l'application ; par exemple si on considère une application audio qui nécessite une fréquence d'échantillonnage de 40 KHz et 100 instructions en série et utilisant un processeur DSP avec un temps de cycles de 30 ns, alors en $3\mu s$ ($30ns \times 100$) le calcul est achevé. Finalement, on peut dire qu'on a une application en temps réel tant que le temps d'attente est supérieur ou égale à zéro (voir figure 5.4).

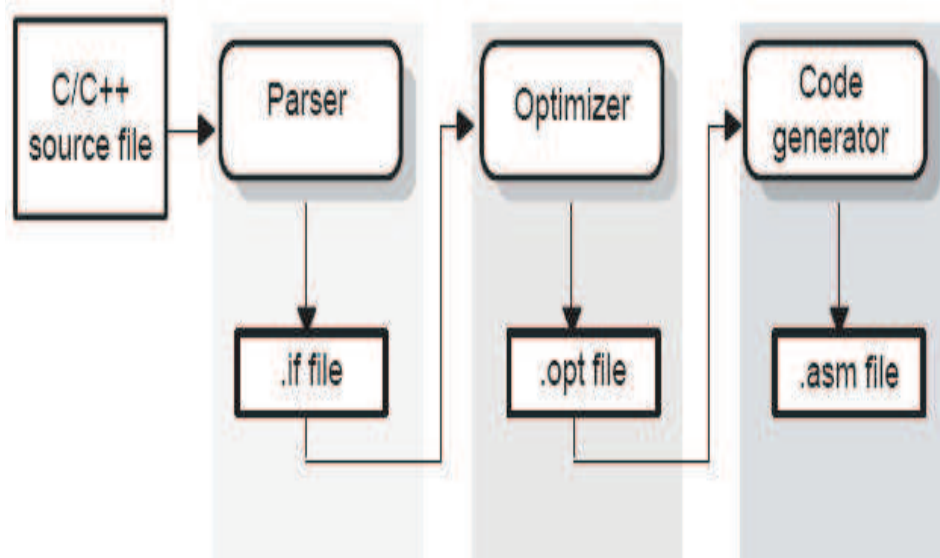


Figure 5.5 — Compilation d'un programme C/C++ avec optimisations.

5.3 Implémentation des algorithmes ASOBI, Robuste-ASOBI et Robuste-k-ASOBI sur carte DSK C6713

Cette partie de la thèse présente quelques résultats issus des implémentations des trois algorithmes choisis (ASOBI, Robuste ASOBI et Robuste k-ASOBI), ces trois algorithmes sont choisis au regard de leurs performances diverses. Cette présentation est effectuée afin de valider nos programmes sur DSP en respectant la contrainte du temps réel.

5.3.1 Évaluation du temps d'exécution et optimisation

Les outils de "Profiling" du "Code Composer" aident à analyser des programmes et permettent d'obtenir les statistiques d'exécution de notre programme en déterminant le nombre moyen de cycles d'horloge.

Quand l'optimisateur est appelé, les étapes suivantes sont exécutées. Un programme en C passe d'abord par un analyseur syntaxique lequel exécute des fonctions de pré-traitement et génère un fichier intermédiaire (.if) qui devient le fichier d'entrée de l'optimisateur. Ce dernier génère par la suite un fichier (.opt) qui passe par le générateur de code pour d'autres optimisations et obtenir ainsi un fichier ASM (figure 5.5).

Les résultats des niveaux d'optimisation sont :

1. (-o0) : optimisation basic.
2. (-o1) : optimisation de l'utilisation des registres.
3. (-o2) : optimisation locale des fonctions en addition à l'optimisation précédente.

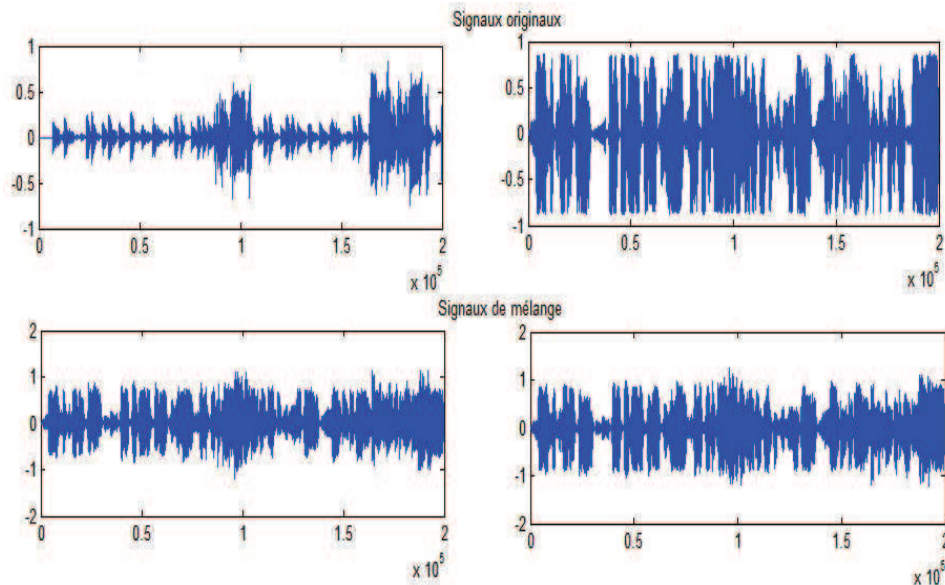


Figure 5.6 — les signaux mélangés x_1 et x_2 à implémenter sur carte DSK.

4. ($-o3$) : optimisation globale en addition des optimisations $-o0$ et $-o1$.
5. ($-o4$) : optimisation du fichier en plus des optimisations précédentes.

5.3.2 Validation des trois algorithmes sur simulateur

Nous mélangeons sous Matlab les signaux sources (originaux) avec la matrice de mélange \mathbf{H} . Le résultat est montré sur la figure 5.6.

L'implémentation des trois algorithmes ASOBI, Robuste-ASOBI et Robuste-k-ASOBI sur la carte DSK (voir la figure 5.7), consiste à :

- Écrire les deux algorithmes en langage C.
- Générer l'exécutable (compilation par le code composer studio).
- Charger les deux vecteurs x_1 et x_2 (échantillonnés) sous code composer, qui sont les signaux mélangés sous Matlab.
- Vérifier les signaux estimés avec les signaux originaux.
- Vérifier la contrainte du temps réel ; que l'on abordera dans le paragraphe suivant.

Le code composer, logiciel de simulation du DSP, offre la possibilité de visualisation des signaux de mélange (Figure 5.8) et les signaux traités à tous les niveaux. Ainsi, nous pouvons voir aux figures 5.9, 5.11 et 5.12 les diagrammes des sources estimées. Nous constatons à partir de cette figure que les signaux sources estimés se confondent avec les signaux originaux de la figure 5.6.

5.3. IMPLÉMENTATION DES ALGORITHMES ASOBI, ROBUSTE-ASOBI ET ROBUSTE-K-ASOBI SUR CARTE DSK C6713

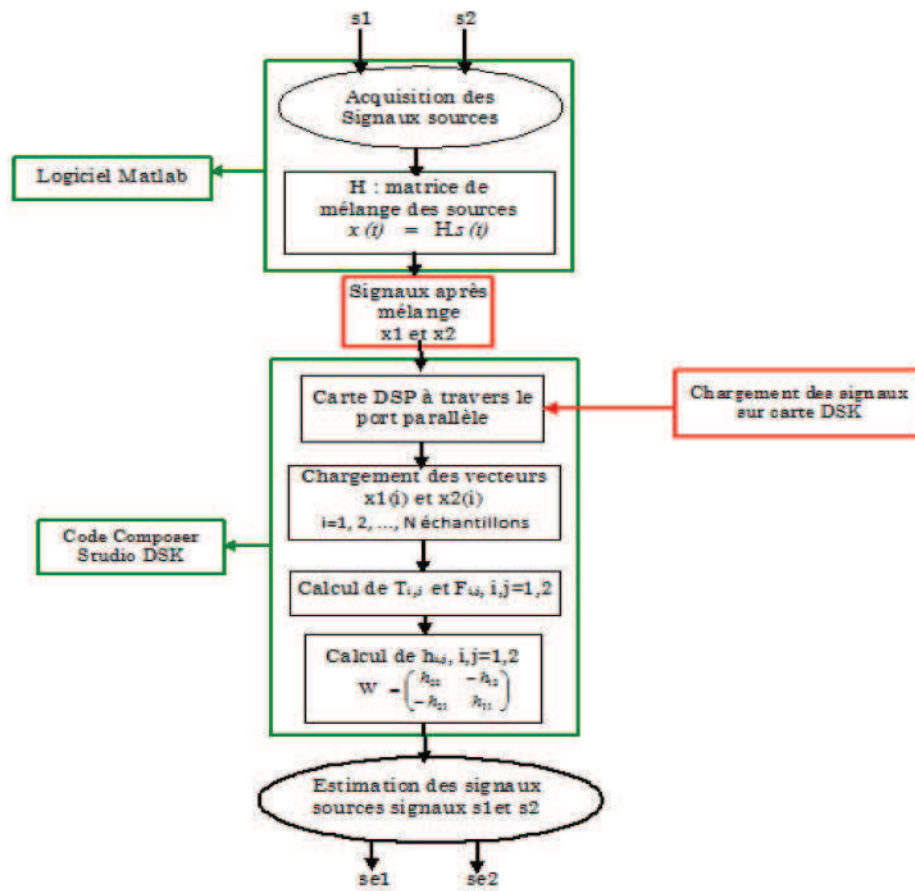


Figure 5.7 — Le diagramme bloc de l'implémentation des algorithmes à implémenter, où se_1 et se_2 sont les signaux sources estimés.

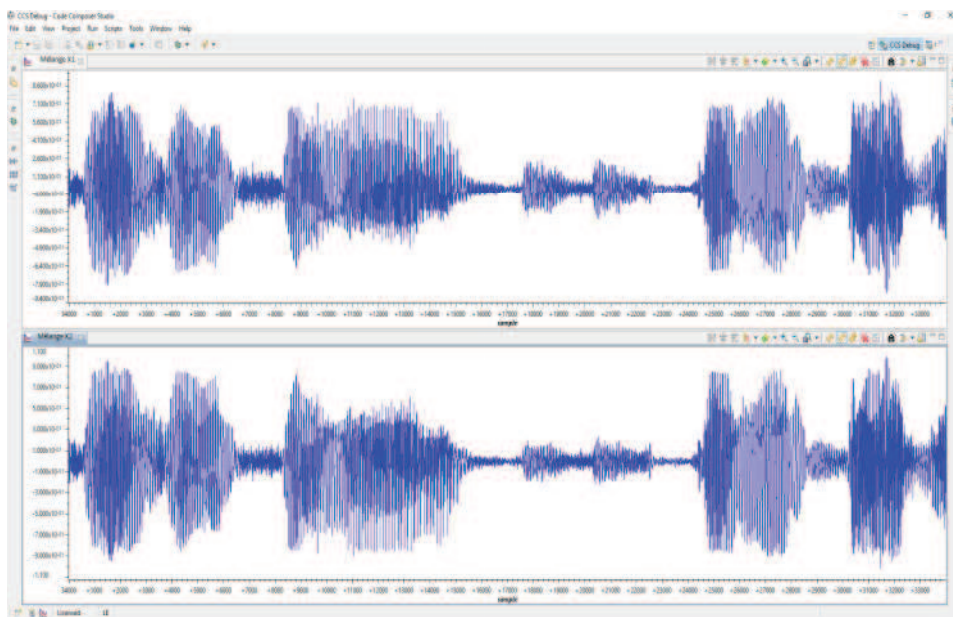


Figure 5.8 — Diagramme des mélanges avant traitement.

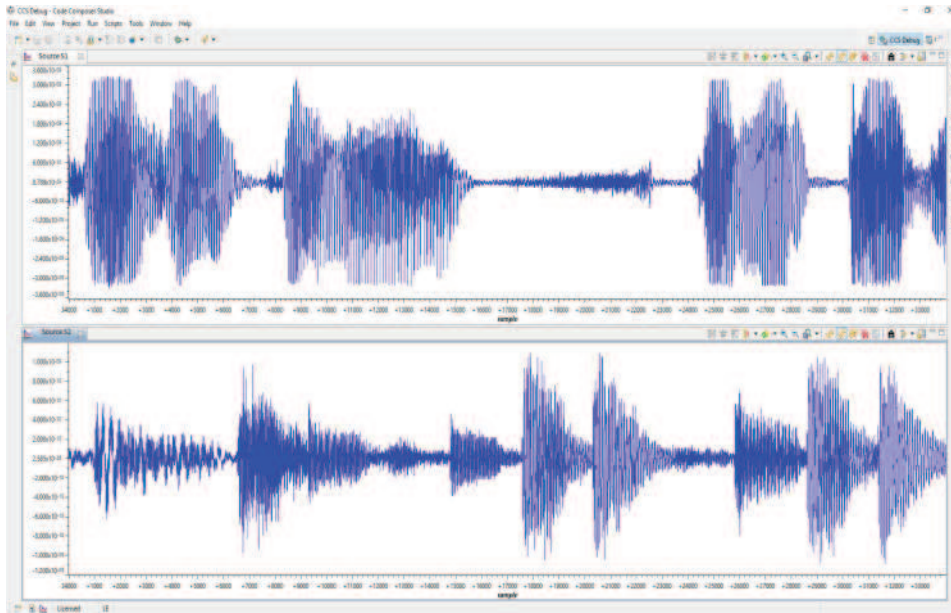


Figure 5.9 — Résultats de traitement après implémentation pour 34000 échantillons avec ASOBI.

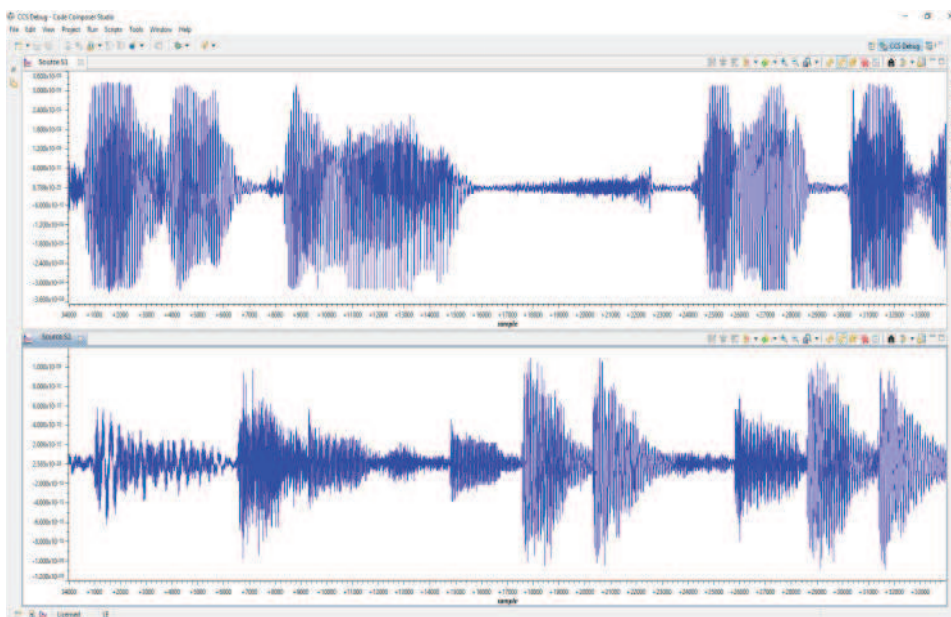


Figure 5.10 — Résultats de traitement après implémentation pour 34000 échantillons avec K-ASOBI.

5.3. IMPLÉMENTATION DES ALGORITHMES ASOBI, ROBUSTE-ASOBI ET ROBUSTE-K-ASOBI SUR CARTE DSK C6713

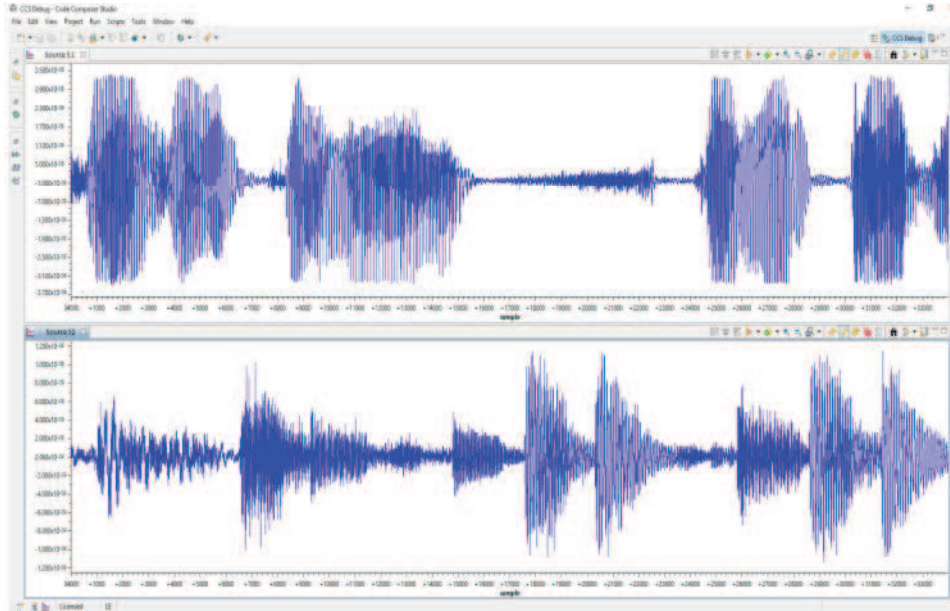


Figure 5.11 — Résultats de traitement après implémentation pour 34000 échantillons avec Robust-ASOBI.

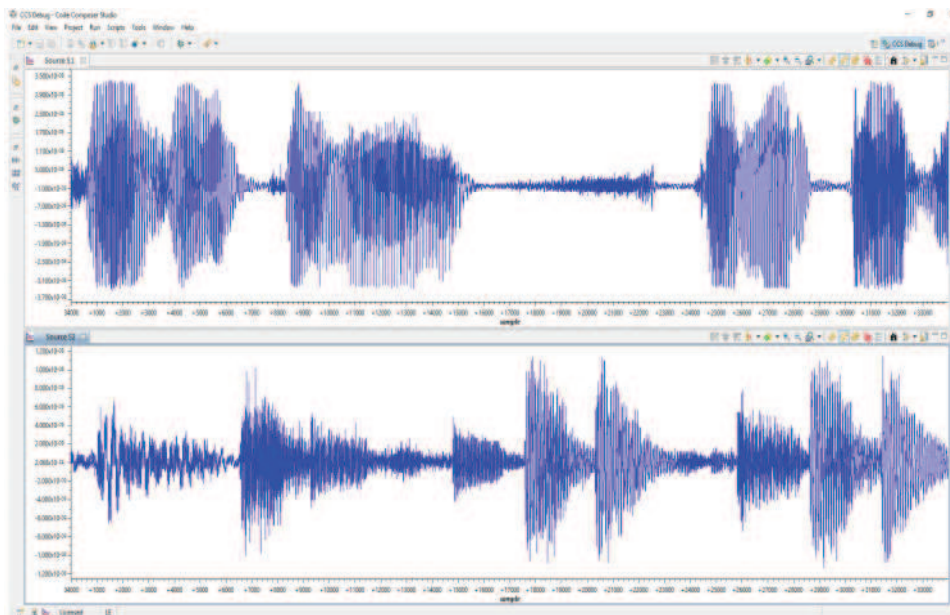


Figure 5.12 — Résultats de traitement après implémentation pour 34000 échantillons avec Robust-k-ASOBI.

5.3.3 Contrainte d'implémentation en temps réel

Pour l'implémentation des trois algorithmes en temps réel, il est nécessaire de choisir un DSP capable d'exécuter les programmes de traitement en temps réel [45]. Dans notre application, la fréquence de prélèvement d'échantillons est de 8000 Hz, le traitement est fait par bloc (une trame de 512 échantillons qui transmettent à 8 octets de données, 8 bits par échantillons), ce qui correspond à un intervalle de 64 ms entre deux trames consécutives. Cet intervalle est le temps maximal tolérable pour l'exécution de l'algorithme ASOBI en temps réel d'un bloc de 8 octets. Cet intervalle correspond à 12800000 cycles d'horloge pour un DSP TMS320C6713 (le cycle est de 5 ns).

Dans le but d'exécuter les tâches de traitement indiqué, il est nécessaire de réduire et d'optimiser la durée d'exécution de notre programme. Pour cela, il est intéressant de savoir quelles sont les parties du programme qui consomment le plus de temps. Nous optimisons alors tout d'abord ces parties du programme. L'évaluation du temps de traitement est exprimée en terme de nombre de cycles. Le temps total sera calculé pour un temps de cycle de 5 ns pour un DSP TMS320C6713.

Pour évaluer le temps de traitement en utilisant le code composer studio, nous spécifions le début et la fin du programme (ou d'une partie d'un programme) par des « Break points » (points d'arrêt), nous initialisons ensuite l'horloge « Clock » ; enfin nous lancerons l'exécution du programme.

Dans notre travail, les contraintes de traitement en temps réel sont gérées de façon à satisfaire les exigences du signal de parole. Plusieurs tests sur les algorithmes ASOBI, Robust-ASOBI et Robust-k-ASOBI ont été effectués avec différentes tailles de paquets d'échantillons (voir les tableaux 5.1, 5.2, 5.3 et 5.4). Le temps d'exécution de notre programme est largement inférieur au temps maximal tolérable qui est d'environ 12800000 cycles. Ainsi, nous pouvons avancer que le traitement se fait en temps réel. Pour les trois algorithmes choisis, le meilleur moment d'exécution est enregistré pour le niveau d'optimisation "fichier". Notons, à cet égard, que le niveau d'optimisation dépend fortement de la structure du programme, l'organisation de la mémoire et l'utilisation des enregistreurs de DSP.

Dans les quatre tableaux, précédents, on peut voir que les quatre algorithmes présente un temps de calcul différents. Nous remarquons aussi que si on passe à 8192 échantillons, le programme ne s'exécute pas, et pour cause, nous avons consommé l'espace mémoire disponible sur la carte DSP malgré que la taille maximal du paquet qu'on peut traiter en temps réel n'a pas été dépassée. Les figures (5.13, 5.14, 5.15 et 5.16) montrent que le temps de calcul dépend fortement du niveau d'optimisation, les niveaux "File" et "Function" permettent de traiter un plus grand nombre d'échantillons avant d'atteindre la limite du traitement liée au temps réel.

Dans les figures (5.17, 5.18, 5.19, 5.20 et 5.21) on compare les quatre algorithmes par rapport au niveau d'optimisation, on remarque que le nombre de cycles d'horloge est similaire pour chacun des quatre algorithmes suivant le niveau.

5.3. IMPLÉMENTATION DES ALGORITHMES ASOBI, ROBUSTE-ASOBI ET ROBUSTE-K-ASOBI SUR CARTE DSK C6713

Temps d'exécution(Cycle)	Nombre d'échantillons			
	512	1024	2048	4096
Level None -o0	116534	231734	462134	922934
Level Register -o1	101581	201933	402637	804045
Level Local -o2	82107	163003	324795	648379
Level Function -o3	6585	11705	21945	42425
Level File -o4	4900	8740	16420	31780

TABLE 5.1 — Le temps d'exécution d'un paquet d'échantillons pour différents niveaux d'optimisation de l'algorithme ASOBI.

Temps d'exécution(Cycle)	Nombre d'échantillons			
	512	1024	2048	4096
Level None -o0	116577	231777	468177	922977
Level Register -o1	101633	201985	402689	804097
Level Local -o2	84685	168141	335053	668877
Level Function -o3	6588	11708	21948	42428
Level File -o4	4904	8744	16424	31784

TABLE 5.2 — Le temps d'exécution d'un paquet d'échantillons pour différents niveaux d'optimisation de l'algorithme k-ASOBI.

Temps d'exécution(Cycle)	Nombre d'échantillons			
	512	1024	2048	4096
Level None -o0	120163	231747	462147	922947
Level Register -o1	100572	201941	402645	804053
Level Local -o2	83653	166085	324803	648387
Level Function -o3	6587	11707	21949	42430
Level File -o4	4896	8736	16420	31780

TABLE 5.3 — Le temps d'exécution d'un paquet d'échantillons pour différents niveaux d'optimisation de l'algorithme Robust-ASOBI.

Temps d'exécution(Cycle)	Nombre d'échantillons			
	512	1024	2048	4096
Level None -o0	120177	238961	476529	923665
Level Register -o1	101128	200968	400648	800008
Level Local -o2	86223	171215	341199	681167
Level Function -o3	6591	11711	21951	42431
Level File -o4	4900	8744	16420	31780

TABLE 5.4 — Le temps d'exécution d'un paquet d'échantillons pour différents niveaux d'optimisation de l'algorithme Robust-k-ASOBI.

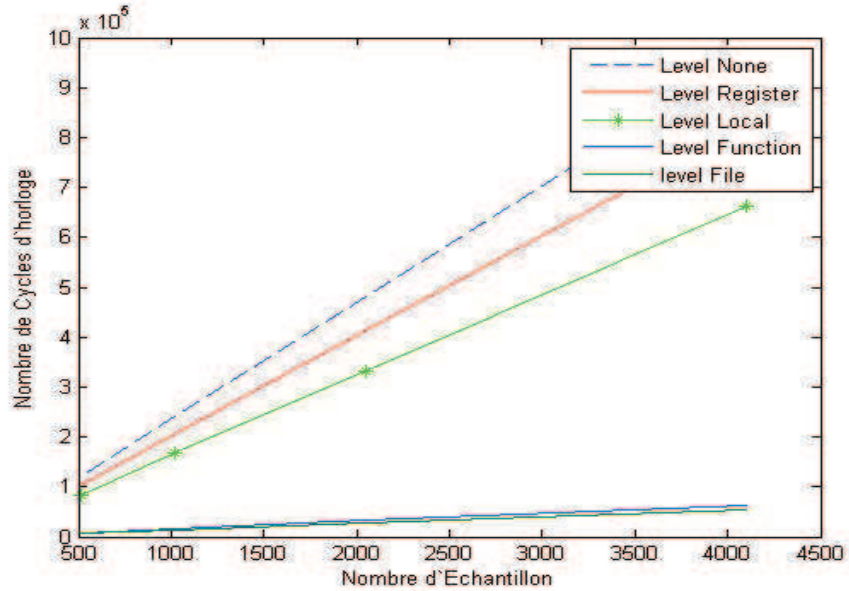


Figure 5.13 — limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme ASOBI.

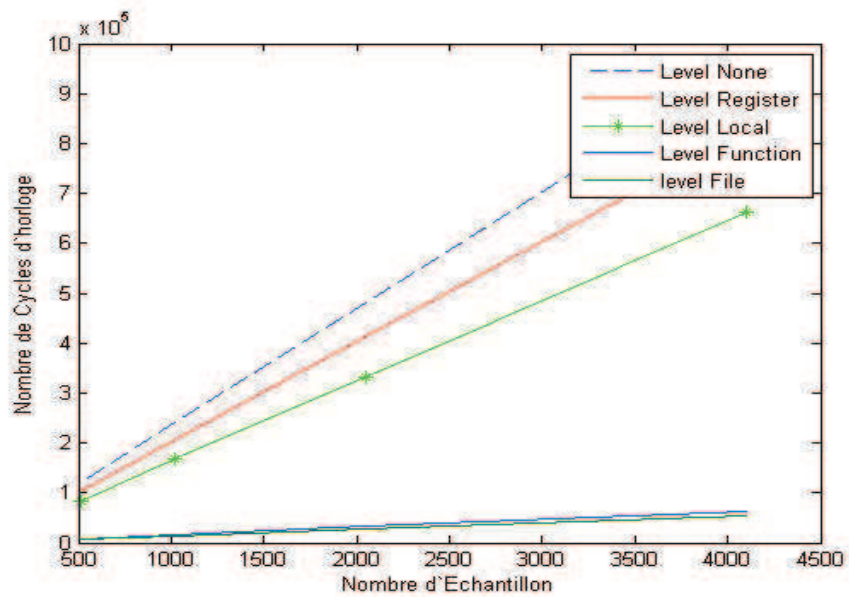


Figure 5.14 — limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme k-ASOBI.

5.3. IMPLÉMENTATION DES ALGORITHMES ASOBI, ROBUSTE-ASOBI ET ROBUSTE-K-ASOBI SUR CARTE DSK C6713

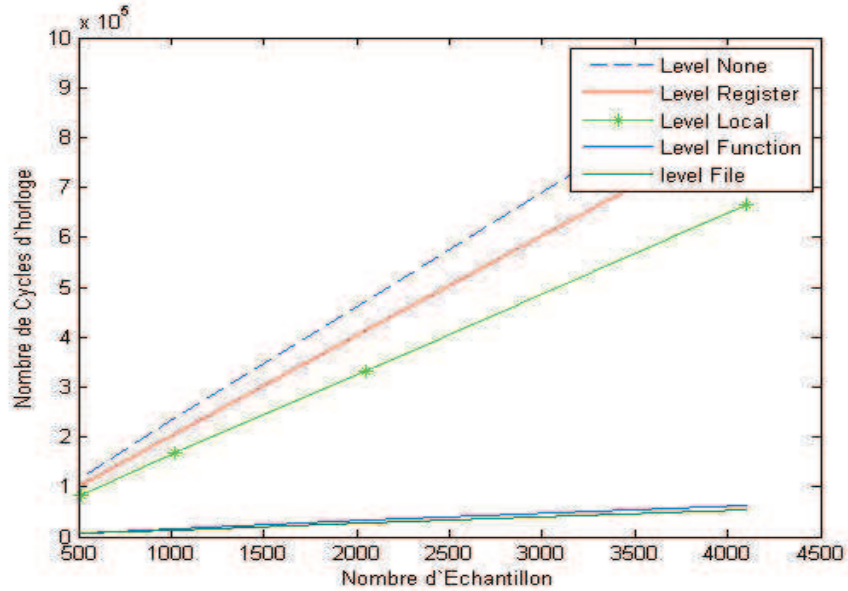


Figure 5.15 — limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme Robust ASOBI.

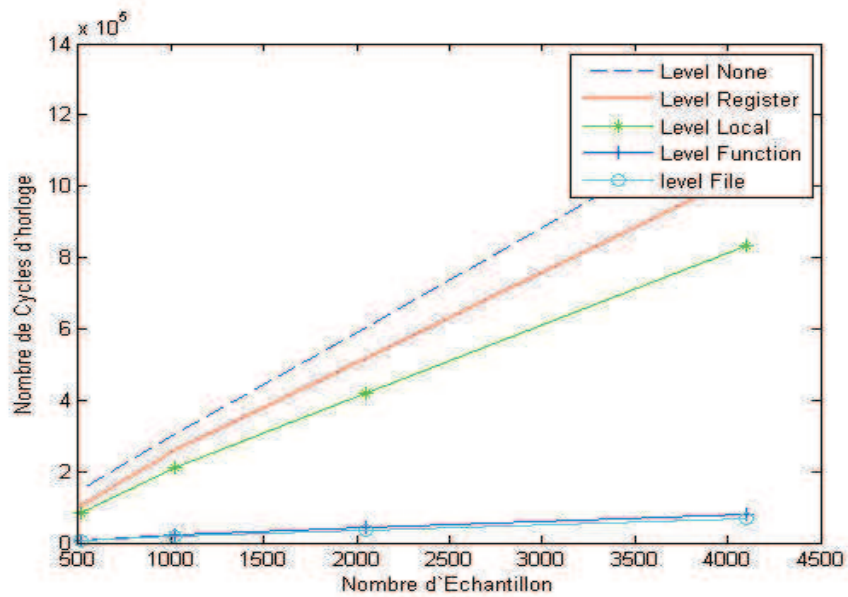


Figure 5.16 — limite du traitement en temps réel en fonction du nombre d'échantillons pour l'algorithme Robust k ASOBI.

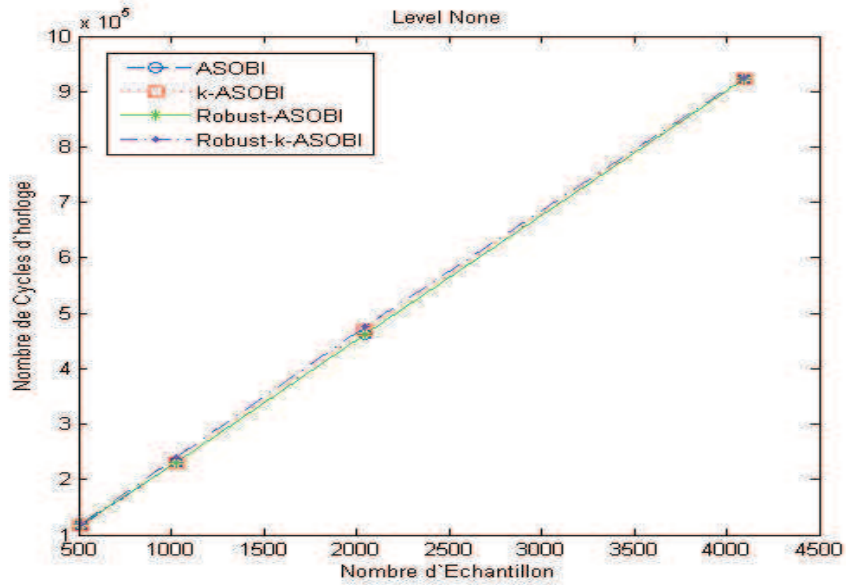


Figure 5.17 — Nombre de cycles d'horloge des quatre algorithmes pour le niveau None.

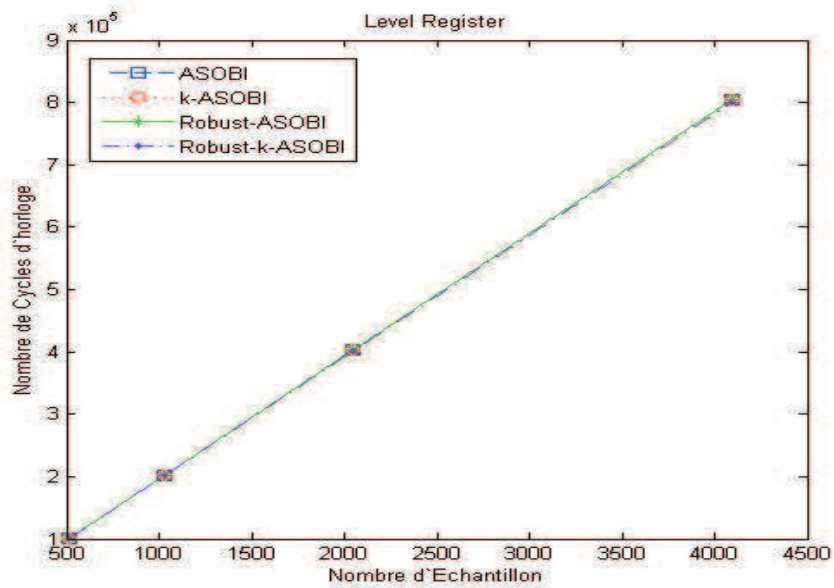


Figure 5.18 — Nombre de cycles d'horloge des quatre algorithmes pour le niveau Register.

5.3. IMPLÉMENTATION DES ALGORITHMES ASOBI, ROBUSTE-ASOBI ET ROBUSTE-K-ASOBI SUR CARTE DSK C6713

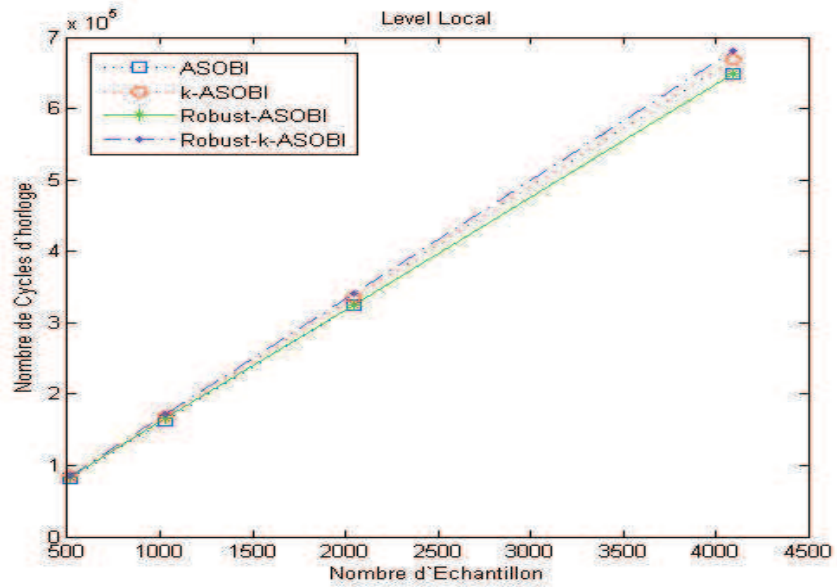


Figure 5.19 — Nombre de cycles d'horloge des quatre algorithmes pour le niveau Local.

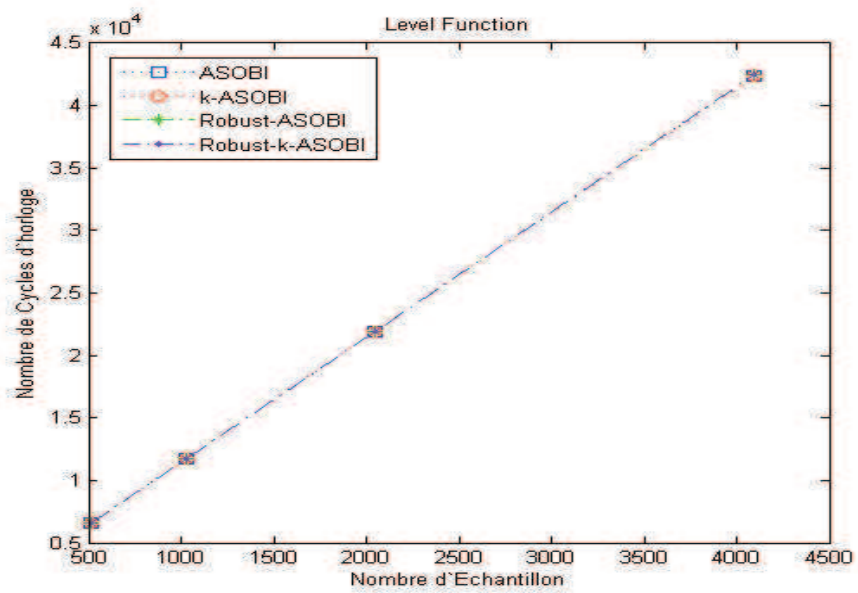


Figure 5.20 — Nombre de cycles d'horloge des quatre algorithmes pour le niveau Function.

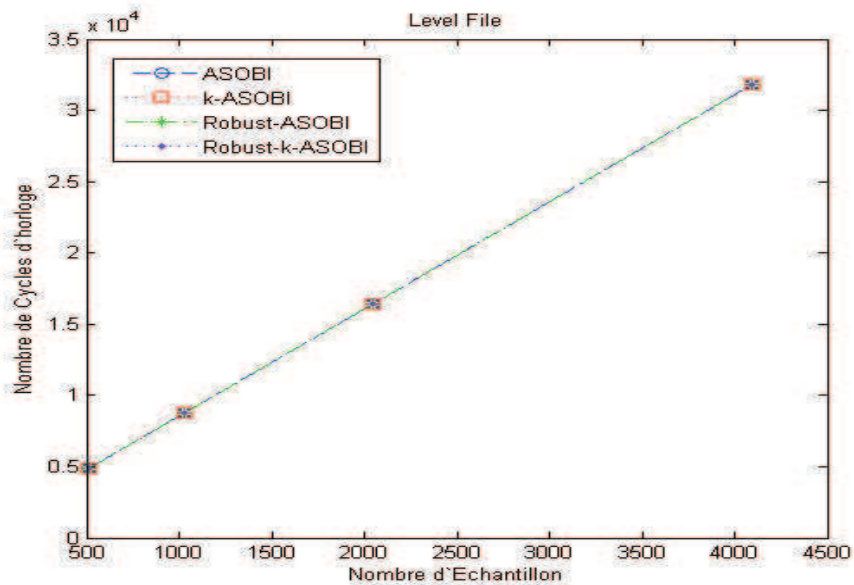


Figure 5.21 — Nombre de cycles d'horloge des quatre algorithmes pour le niveau File.

5.4 Conclusion

Dans ce chapitre, nous avons présenté l'implémentation en temps réel de notre algorithme ASOBI sur carte DSP de la famille C6000 de Texas Instruments. En particulier le processeur TMS320C6713 qui est le coeur de la carte DSK (support de développement). Nous avons proposé des améliorations sur le temps d'exécution de l'algorithme en exploitant les différents niveaux d'optimisation qu'offre le logiciel Code Composer Studio. Nous avons aussi validé notre implémentation DSP de trois autres algorithmes k-ASOBI, Robust-ASOBI et Robust-k-ASOBI et vérifié les conditions d'implémentation de ces quatre algorithmes en temps réel. L'implémentation sur DSP a montré que les quatre algorithmes présentent des temps de calcul similaires qui dépend fortement du niveau d'optimisation choisi.

Conclusion Générale

Dans ce travail nous avons considéré le problème de la séparation de sources, en particulier nous nous sommes intéressés au problème de la séparation de mélanges linéaires instantanés de sources indépendantes. Ceci reste un cas typique en traitement d'antenne où les algorithmes classiques nécessitent la connaissance a priori du modèle physique de propagation ainsi que la géométrie de l'antenne. La séparation de sources quant à elle, fournit une approche non paramétrique pour le traitement d'antenne indépendamment de toute modélisation physique. Les approches que nous avons considérées exploitent des informations statistiques différentes sur les signaux source.

La résolution de ce problème revient souvent à celui de la recherche de composantes indépendantes dans un mélange linéaire. Par ailleurs une identification complète de la matrice de mélange est impossible dans un contexte aveugle. La matrice de mélange ne peut être estimée qu'à un facteur et une permutation près de ses colonnes.

Certaines méthodes de séparation aveugle de sources procèdent d'abord par l'identification aveugle du mélange ensuite restituent les signaux sources par des procédures de filtrage spatial. Dans un contexte aveugle l'hypothèse de base en séparation de source est l'indépendance statistique des signaux sources. Cette hypothèse ne suffit à elle seule pour résoudre le problème de la séparation de sources. Il est nécessaire alors de disposer d'une hypothèse supplémentaire.

Dans cette optique, le travail a porté sur la séparation de deux sources à partir de signaux reçus par deux capteurs. Cette thèse a ainsi concerné le développement de nouveaux algorithmes de séparation de source à coût de calcul réduit dédiés à des applications temps-réel et des implémentations hardwares. Nous avons alors proposé plusieurs algorithmes remplissant cet objectif.

Pour cela, nous avons proposé d'abord une première classe d'algorithmes qui coïncident avec le cas de bruit additif spatialement blanc. On peut citer alors les algorithmes ASOBI, k-ASOBI et Optimal k-ASOBI. Ces derniers exploitent les corrélations des signaux reçus ce qui nécessite la connaissance ou l'estimation de la variance du bruit. Pour éviter ce calcul de l'estimée de la variance du bruit, nous avons proposé ensuite une deuxième classe d'algorithmes qui n'exploitent que les corrélations des signaux à des retards différents de zéro. Ces derniers sont alors exploitables même dans le cas de bruit spatialement corrélé pourvu qu'ils soient temporellement blancs.

L'identifiabilité des algorithmes proposés a été étudiée et les conditions pour lesquelles les coefficients de la matrice de mélange peuvent être identifiables ont été dé-

duites. Celles-ci sont fonctions des statistiques utilisées par chaque algorithme proposé. Si les signaux sources présentent des spectres normalisés identiques, les algorithmes proposés ne permettent pas de les séparer. Il faudra alors avoir recours aux statistiques d'ordre supérieur à deux.

Une des qualités des algorithmes proposés est leur faible coût de calcul. Ces algorithmes ne nécessitent pas d'opérations de division, ce qui rend leur implémentation hardware (sur circuits FPGA ou DSP) plus aisées. Ces opérations, dont l'implémentation est onéreuse, ont été évitées grâce à l'exploitation du concept de l'identification aveugle qui stipule que le recouvrement des signaux source ne peut se faire qu'à un facteur et une permutation près. Un autre avantage majeur des algorithmes développés dans cette thèse est le fait qu'ils soient analytiques, c'est à dire qu'ils fournissent les estimés en question (matrice de mélange, signaux sources) par un calcul direct sans avoir recours à aucune procédure itérative et donc évitent tout problème de convergence.

Il a été proposé dans cette thèse, aussi, des versions adaptatives de deux algorithmes développés, en l'occurrence ASOBI et k-ASOBI. Celles-ci peuvent permettre de considérer des environnements variants dans le temps. L'étude et la comparaison des performances des algorithmes proposés de second ordre ont montré que nos algorithmes présentent de bonnes performances. En plus ces algorithmes offrent une simplicité d'implémentation sur carte DSP, puisque ils ne possèdent aucune division.

Le processeur numérique du signal TMS320C6713 allie à la fois la flexibilité et la puissance de calcul pour la mise en œuvre des configurations d'implémentation proposées tout en respectant les contraintes imposées par les quatre algorithmes choisis, en l'occurrence ASOBI, Robust-ASOBI, k-ASOBI et Robust-k-ASOBI. Nous pouvons ainsi dire, que nos algorithmes sont implémentables et l'évaluation du temps d'exécution a montré la faisabilité de traitement en temps réel.

Les perspectives dans le domaine de la séparation de sources, semblent être illimitées, compte tenu, d'une part, des applications que propose la séparation de sources et des différents modèles à traiter (exemple : cas où on a plus de sources que de capteurs), d'autre part, du développement exponentiel de la vitesse d'exécution des processeurs de traitement numérique du signal. Néanmoins, comme perspectives proches, on peut citer :

- Implémenter la version adaptative de ASOBI et ses versions sur carte DSP ;
- Extension de la solution analytique du second ordre au Cas de signaux à valeurs Complexes ;
- Analyse théorique des performances en terme de taux de réjection moyen ;
- Proposer une solution analytiques pour trois sources et autant de capteurs ;
- Réalisation d'un banc d'essai pour séparer deux sources émettrices à travers deux récepteurs RF (Radio- Fréquence) raccordés à l'entrée de la carte DSP.

Références

- [1] Sophie ACHARD. *Mesures de dépendance pour la séparation aveugle de sources*. PhD thesis, Université JOSEPH FOURIER (GRENOBLE I), France, 2008.
- [2] Comon Pierre and Christian. Jutten. *Handbook of Blind Source Separation, Independent Component, Analysis and Applications*. Elseiver, 2010.
- [3] A. Meziane Bentahar Meziane, A. Belouchrani, and E. Bourennane. An analytical solution for a second order blind identification algorithm. Guelma, Algeria, CNTSA 2005 19 et 20 septembre 2005.
- [4] A. Fermas, A. Belouchrani, and O. Aitmohamed. Hardware implementation of free division block-based bss algorithm. In *Circuits and Systems and TAISA Conference, 2009. NEWCAS-TAISA '09. Joint IEEE North-East Workshop on*, pages 1–4, 2009.
- [5] S. Haykin. *Adaptive filter theory*. New Jersey, Prentice Hall, Englewood Cliffs, 1986.
- [6] B. Kacha, K. Abed-Meraim, and A. Belouchrani. Fast Adaptive Blind MMSE Equalizer for Multichannel FIR Systems. In *EURASIP Journal on Applied Signal Processing*, volume 2006, 17 pages 2006.
- [7] A. Hypvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.
- [8] Christian Jutten and Jeanny Herault. Blind separation of sources, part 1 : An adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 24(1) :1–10, aug 1991.
- [9] Christian Jutten and Taleb. Anisse. Source separation from dusk till dawn. *Independent Component Analysis*, pages 15–26, June 2000.
- [10] J F Cardoso. The three easy routes to independent component analysis ; contrasts and geometry. *Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA01)*, pages 1–6, 2001.
- [11] Anthony Larue. *Séparation de sources markoviennes*. PhD thesis, faculté polytechnique de Mons, France, juin 2003.
- [12] Comon. Pierre. Independent component analysis, a new concept? *Signal Processing*, pages 3(36) :287 (314),, April 1994.
- [13] G. Darmois. Analyse générale des liaisons stochastiques : étude particulière de l'analyse factorielle linéaire. *Revue de l'Institut International de Statistique / Review of the International Statistical Institute*, 21(1/2) :2–8, 1993.
- [14] J F Cardoso. Statistical principles of source separation. In *IFAC-1997*, 1997.
- [15] A. Belouchrani and Abed-Meraim . Séparation aveugle au second ordre de sources temporellement corrélées. In *In 14eme colloque GRETSI, Juan-les-Pins, France*, page 309–312, Sep 1993.

- [16] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *Signal Processing, IEEE Transactions on*, 45(2) :434–444, feb 1997.
- [17] A. Taleb and C. Jutten. Sources separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing.*, pages 10–47, oct 1999.
- [18] Adel. Belouchrani. *Séparation Autodidacte de Sources : Algorithmes, Performances et Application à des Signaux Expérimentaux*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, 1995.
- [19] A. Aissa-El-Bey, K. Abed-Meraim, and Y. Grenier. Underdetermined Blind Audio Source Separation Using Modal Decomposition. In *EURASIP Journal on Audio, Speech, and Music Processing*, volume 2007, 15 pages 2007.
- [20] A. Kachenoura, L. Albera, L. Senhadji, and K. Bakhru. Séparation Aveugle de Sources en Ingénierie Biomédicale. *Elsevier Masson. on Science Direct*, ITBM-RBM 28, 2007.
- [21] Abdeldjalil Aissa-El-Bey. *Séparation aveugle de Sources Audio*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2007.
- [22] Ali Mansour. *Contribution à la séparation aveugle de sources*. PhD thesis, Grenoble, INPG, Grenoble, 1997. Th. : signal, image, parole.
- [23] N. Linh-Trung, A. Belouchrani, K. Abed-Meraim, and B. Boashash. Separating More Sources Than Sensors Using Time-Frequency Distributions. In *EURASIP Journal on Applied Signal Processing*, volume 2005, pages 2828–2847, 2005.
- [24] J.-P. Deschamps and G. Sutler. Finite field division implementation. In *Field Programmable Logic and Applications, 2005. International Conference on*, pages 670–674, aug. 2005.
- [25] A.E. Bashagha. Novel radix-2k square root module. *Circuits, Devices and Systems, IEE Proceedings -*, 148(4) :190–196, aug 2001.
- [26] Monzingo . Robert A, Haupt . Randy L, and Miller. Thomas W. *Introduction to Adaptive Arrays*. SciTech Publishing, 2011.
- [27] A. Belouchrani, E. Bourennane, and K. Abed-Meraim. A closed form solution for the blind separation of two sources from two sensors using second order statistics. In *EUSIPCO-2006*.
- [28] E. Moreau and O. Macchi. Ieee sp workshop on higher order stat, lake tahoe, usa. In *Lake Tahoe, USA 1993*, 1993.
- [29] D. T.Pham, P.Garat, and C Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. In *EUSIPCO*, pp.771-774, 1992.
- [30] Hicham Ghennioui. *Séparation aveugle de mélanges linéaires convolutifs de sources corrélées*. PhD thesis, co-tutelle Université Med V-Agdal, Maroc et Université du Sud-Toulon, France, 2008.
- [31] Ouedraogo Wendyam. *Geometrical method for non-negative source separation : Application to dynamic PET imaging and mass spectrometry*. PhD thesis, Université Grenoble Alpes, France, Novembre 2012.
- [32] Jaureguiberry Xavier. *Fusion for audio source separation*. PhD thesis, Télécom Paris-Tech, France, Juin 2015.
- [33] Victor Maurandi. *Algorithmes pour la diagonalisation conjointe de tenseurs sans contrainte unitaire. Application à la séparation MIMO de sources de télécommunications numériques*. PhD thesis, Université de Toulon, France, Novembre 2015.

- [34] Becker Hanna. *Denoising, separation and localization of EEG sources in the context of epilepsy*. PhD thesis, Université Nice Sophia Antipolis, France, Octobre 2014.
- [35] V. Zagorodnov, A. Hakim A. Laghrib A. Mamouni N Ciptadi, A. Ghazdali, and Said Raghay. A new method for the extraction of fetal ecg from the dependent abdominal signals using blind source separation and adaptive noise cancellation techniques. *Theoretical Biology and Medical Modelling*, 30(10.1186/s12976-015-0021-2) :20, November 2015.
- [36] H. Gaoming, L.Yang, and G.Su. Blind source separation used for radar anti-jamming. In *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, volume 2, pages 1382–1385 Vol.2, Dec 2003.
- [37] D. Yuling and Z. Hang. Noisy blind signal-jamming separation algorithm based on vbica. *Wireless Personal Communications*, 74(2) :307–324, 2014.
- [38] Fulei.Chu Zhinong.Li, Yongyong.He. Application of the blind source separation in machine fault diagnosis : A review and prospect. In *Mechanical Systems and Signal Processing*, 2005.
- [39] Yasmina BENABDERRAHMANE. *Séparation aveugle de signaux de parole utilisant les Statistiques d'ordre supérieur et la décomposition en sous-espaces*. PhD thesis, Université du Quebec, Canada, 2011.
- [40] D. Torrieri and K. Bakhru. Direction finding for spread-spectrum systems with adaptive arrays. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7, oct. 2006.
- [41] A. Meziane Bentahar Meziane, A Aissa-El-Bey, T. Chonavel, and A. Belouchrani. Une solution analytique pour l'identification aveugle au second ordre d'un système à deux entrées et deux sorties : implémentation adaptative. 4/2016, 1-17(4) :190–196, Dec 2016.
- [42] A. Meziane Bentahar Meziane, B. Magaz, and A. Belouchrani. A DSP Implementation of a Second Order Blind Identification Algorithm Using TMS320C6711 Processor. In *Proc. of DAT06, 3^{eme} Séminaire sur les Systemes de Detection Architectures et Technologie*, Algiers, Algeria, 20 au 22 Novembre 2006.
- [43] E. SICARD et S. Delmas-Bendia. *Une introduction aux DSP*. Proposition au bulletin de l'Union des Physiciens, 2001.
- [44] M. Torres. *Contribution à l'étude de l'implantation d'algorithmes de traitement de signal sur plateformes DSP-DSP et DSP-FPGA*. Thèse de doctorat, Université de Limoge, France, Dec 2003.
- [45] R. Chassaing. *DSP Application Using C and the TMS320C6x DSK*. Edition Wiley, 2002.