Sur quelques applications du codage parcimonieux et sa mise en oeuvre

Bertrand Coppa

UNIVERSITÉ DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : Signal, Image, Parole, Télécoms

Arrêté ministériel : 7 août 2006

Présentée par

Bertrand COPPA

Thèse dirigée par **Olivier MICHEL**

préparée au sein du **CEA-Léti**

dans l'**École Doctorale : EEATS**

Sur quelques applications du codage parcimonieux et sa mise en œuvre.

Thèse soutenue publiquement le **8 mars 2013**,

devant le jury composé de :

M. André FERRARI

Professeur, Université de Nice - Sophia Antipolis, Rapporteur

M. Nicolas DOBIGEON

Maître de conférence HDR, Université de Toulouse, Rapporteur

M. Pierre BORGNAT

Chargé de recherche, ENS Lyon, Examinateur

M. Jérôme MARS

Professeur, Université de Grenoble, Examinateur

M. Olivier MICHEL

Professeur, Université de Grenoble, Directeur de thèse

M. Dominique DAVID

Ingénieur-chercheur HDR, CEA-Léti, Encadrant

M. Éric MOISAN

Maître de conférence, Université de Grenoble, Invité

Remerciements

Au cours de mes travaux de thèse, j'ai côtoyé de nombreuses personnes auxquelles je voudrais dédier cette page.

Pour commencer, je remercie Olivier Michel, mon directeur de thèse, pour l'encadrement, ses conseils et sa patience tout au long de ce projet. Je remercie également Éric Moisan pour sa collaboration et surtout, son aide très précieuse lors de la rédaction de ce document. Je pense aussi à Boris Kouassi et Manel Zidi, avec qui j'ai travaillé lors de leur stage de Master. Merci aussi à mon encadrant Dominique David, ainsi qu'à Rodolphe Héliot grâce à qui je me suis intéressé aux problèmes des signaux neuronaux. J'ai une pensée aussi pour Jean-Louis Lacoume, sa science et son enthousiasme lors de chacune de nos rencontres.

Je voudrais remercier tous les collègues doctorants et docteurs : Anne-Cécile, Mikael, Mathieu, Rim, Seifeddine, Nawres, Régis, ... Nous avons partagé ensemble les moments de doutes, les difficultés, ainsi que les réjouissances de la thèse. Je pense aussi aux collègues de bureau : Jean-Charles, Nathalie, Patrick, Laurent, Audrey, Malvina, Viviane, Christelle, François, et les autres avec qui j'ai échangé au cours de ces 3 années passées au LETI.

Je voudrais aussi remercier ma famille, Charlotte, mes amis (Maxime et Grégory, qui travaillaient eux-aussi sur une thèse, Cécile, ...) pour leur soutien mais surtout pour le temps passé ensemble.

Enfin, un dernier merci aux rapporteurs et autres membres du jury qui ont accepté de juger mon travail.

Bertrand

Table des matières

Liste des notations

Liste des acronymes

- **CAI** : Convertisseur Analogique Information ;
- **CS** : Compressed Sensing ;
- **IRLS** : Iteratively Reweighted Least Squares ;
- **MP** : Matching Pursuit ;
- **OMP** : Orthogonal Mathing Pursuit ;
- **FOCUSS** : Focal Underdetermined System ;
- **RIP** : *Restricted Isometry Property*, propriété d'isométrie restreinte ;

Notations

Soient x et y deux vecteurs de $\mathbb{R}^N$, $x = [x_1, x_2, ..., x_N]^T$ et $y = [y_1, y_2, ..., y_N]^T$. On notera le produit scalaire

$$< x, y >= x^T y = \sum_{i=1}^{N} x_i y_i$$

et $\|x\|_2^2 =< x, x >$, la norme associée au produit scalaire.

Pour rappel, on redonne ici les définitions d'une norme et celle d'une distance :

Définition 1 (Norme) *Soit $\mathbb{K}$ un corps commutatif muni d'une valeur absolue et $\mathbf{E}$ un $\mathbb{K}$-espace vectoriel. Une norme sur $\mathbf{E}$ est une application $\mathcal{N}$ de $\mathbf{E}$ à valeur réelle positive satisfaisant les propriétés suivantes :*

- ***séparation :*** $\forall x \in \mathbf{E}$, $\mathcal{N}(x) = 0 \Leftrightarrow x = 0_{\mathbf{E}}$;
- ***homogénéité :*** $\forall (\lambda, x) \in \mathbb{K} \times \mathbf{E}$, $\mathcal{N}(\lambda \cdot x) = |\lambda| \cdot \mathcal{N}(x)$;
- ***sous-additivité :*** $\forall (x, y) \in \mathbf{E}^2$, $\mathcal{N}(x + y) \leq \mathcal{N}(x) + \mathcal{N}(y)$;

Définition 2 (Distance) *Une distance d sur un ensemble $\mathbf{E}$ est une fonction de $\mathbf{E}$ dans $\mathbb{R}$:*

$$d \ : \ \mathbf{E} \times \mathbf{E} \to \mathbb{R},$$

qui satisfait les conditions suivantes pour tout x, y et z de $\mathbf{E}$:

- ***identité des indiscernables :*** $d(x, y) = 0 \Leftrightarrow x = y$;
- ***symétrie :*** $d(x, y) = d(y, x)$;
- ***sous-additivité :*** $d(x, y) \leq d(x, z) + d(z, y)$;

Les propriétés d'identité des indiscernables et de sous-additivité entrainent $d(x, y) \geq 0$.

On remarque que si $\mathcal{N}$ est une norme, alors $\mathcal{N}(x-y)$ est une distance.

Pour les matrices, on utilise aussi la norme de Frobenius :

Définition 3 *Soit une matrice A de dimension $N \times M$. La norme de Frobenius de A, notée $\|A\|_F$, est définie de la manière suivante :*

$$\|A\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} |a_{i,j}|^2} = \sqrt{\mathrm{tr}\,(A^*A)} = \sqrt{\mathrm{tr}\,(AA^*)}$$

Chapitre 1

Contexte

La tendance actuelle est à l'instrumentalisation massive à l'aide de nombreux capteurs, dans tous les domaines. Les capteurs se multiplient dans les automobiles, les ouvrages d'art, le corps humain et les applications sont multiples, de la sécurité (automobiles, état des structures dans les ouvrages d'art, surveillance des personnes âgées, ...) à la santé (rééducation, surveillance, ...) ou au sport (analyse des mouvements, des paramètres physiologiques, ...). Parmi ces nombreuses applications, nous nous intéressons dans ce document à deux d'entre elles : la surveillance dans les bâtiments à l'aide d'un réseau de capteurs micro-sismiques, et les signaux neuronaux enregistrés par des capteurs implantés à l'intérieur du cerveau. Dans ces deux situations, l'acquisition des signaux et leur numérisation n'est pas un obstacle vis-à-vis des technologies actuelles. Cependant, si l'on envisage des réseaux de capteurs à grande échelle, les débits numériques peuvent devenir très ou trop élevés (en effet, rien qu'une centaine de capteurs enregistrant à 100 kHz sur 32 bits nécessite une bande-passante de l'ordre de 320 Mbps) et cela pose deux problèmes :

- L'infrastructure nécessaire pour transmettre ces données. Dans le cas de transferts filaires, il faut, outre les câbles, un système pouvant accepter toutes ces connexions et les traiter. Dans le cas de communications sans fil, le spectre disponible et le coût énergétique limitent le débit. De plus, le béton n'est pas un milieu favorable aux transmissions sans fil, pas plus que le cerveau humain.
- La consommation électrique nécessaire à la transmission. La tendance est aux capteurs autonomes. Par autonomes, on entend que les capteurs doivent subvenir à leurs besoins énergétiques à l'aide d'une réserve d'énergie de faible capacité, ou éventuellement une source d'énergie limitée (photovoltaïque, récupération d'énergie mécanique.) Pour réduire le coût énergétique de la transmission, il faut réduire la portée et/ou la quantité d'information à transmettre.

Il est donc opportun d'envisager une compression à même le capteur, pour réduire la quantité de données à transmettre. Cette compression doit alors être simple, sinon le gain du point de vue énergétique obtenu sur la quantité de données à transmettre est perdu par l'opération d'encodage. Cette compression doit permettre de retrouver un signal, tel celui

présenté sur la figure 1.1 sans perdre d'information.

Beaucoup de méthodes de compression, que l'on pourrait qualifier de classiques au sens où on les retrouve dans des applications utilisées quotidiennement par tous, s'appuient sur une représentation du signal qui concentre l'énergie sur quelques coefficients, après avoir échantillonné le signal de manière traditionnelle. Par exemple, on encode la musique à l'aide de MP3 [Shl94, Bra99], qui transforme le signal dans le domaine fréquentiel, pour se défaire de l'information redondante. Les images sont encodées par la méthode JPEG [Wal91], qui s'appuie aussi, entre autres, sur l'information fréquentielle. Ces méthodes ne sont pas sans perte, mais celles-ci peuvent être très faibles tout en permettant une forte compression, parce que ce qui est "jeté" ne contient que peu d'énergie.

Cependant, ces méthodes de compression ont l'inconvénient de nécessiter des calculs complexes, puisqu'il faut souvent effectuer des transformations mathématiques, avant de choisir quels coefficients de représentation conserver. Mais puisqu'on jette une grande partie des échantillons déjà acquis, pourquoi ne pas acquérir uniquement ce dont on a besoin ?

C'est de cette question que part l'idée du codage parcimonieux, en anglais *Compressed Sensing* (CS) : proposer une méthode d'acquisition/compression simultanée qui ne conserve que le nécessaire pour reconstruire le signal. L'échantillonnage traditionnel n'est que peu restrictif sur la nature du signal considéré, si ce n'est qu'il doit être à bande limitée. Mais les méthodes comme JPEG [Wal91, SCE01] ou MP3 [Shl94, Bra99] s'appuient sur des hypothèses plus fortes, notamment qu'il existe une transformation qui permet de concentrer l'énergie du signal, c'est-à-dire l'information pertinente contenue par celui-ci, sur seulement quelques éléments. Ce n'est d'ailleurs pas une idée récente, puisque la transformée de

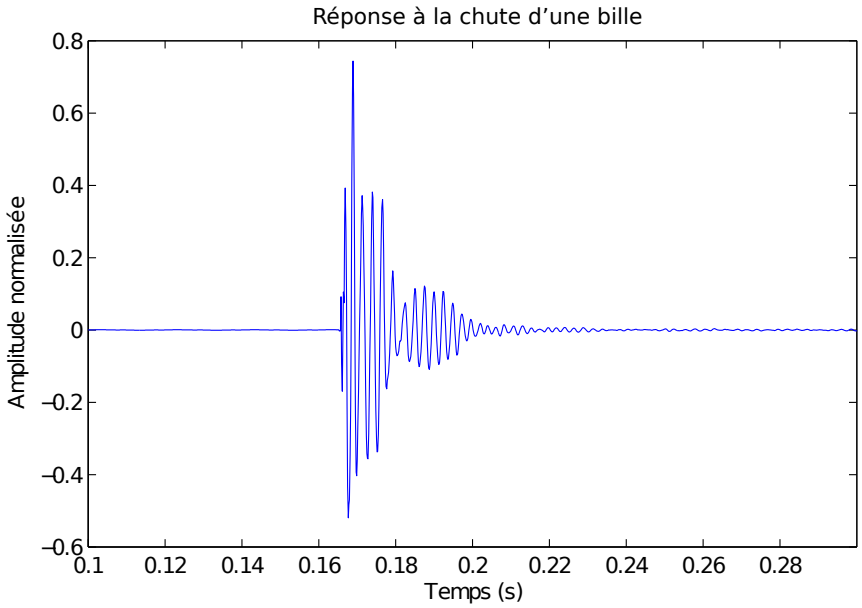

FIGURE 1.1 – Exemple de réponse d'un capteur accéléromètre piezzo-électrique à la chute d'une bille sur une dalle de béton, échantillonnée à $F = 6,4\text{kHz}$

Fourier répondait déjà à ce besoin de représentation parcimonieuse en représentant une sinusoïde à l'aide de 2 coefficients complexes.

L'idée du codage parcimonieux est de faire cet échantillonnage compressé en projetant le signal sur une série de séquences prédéterminées, les résultats de ces projections faisant office d'échantillons. Cette méthode, en plus de réduire le nombre d'échantillons, a l'avantage d'une relative facilité de mise en œuvre : les opérations sont linéaires, et ne requièrent pas de calculs mathématiques complexes. Cependant, cette simplicité a un coût : celui de la reconstruction du signal. Les méthodes de compression usuelles sont coûteuses (notamment parce qu'elles nécessitent des transformations mathématiques), mais la décompression se fait rapidement. Par exemple, un terminal mobile, comme un téléphone, est capable de lire des vidéos de haute qualité, qui nécessitent un temps de compression très long (supérieur à la durée de la vidéo) sur une station de travail. Dans le cas du codage parcimonieux, les rôles sont inversés : c'est la reconstruction qui est coûteuse. En effet, l'information a priori sur le signal est que l'on peut le représenter de façon parcimonieuse, dans une base que l'on connait, mais contrairement aux techniques classiques, on n'acquiert pas tout le signal afin de déterminer les coefficients significatifs et leur valeurs, mais seulement une multiplication par une matrice donnée. L'algorithme doit alors déterminer, à partir du résultat de cette projection, quel mélange de coefficients était présent au départ.

1.1 Illustration du principe du codage parcimonieux

Pour illustrer l'idée du codage parcimonieux, considérons l'exemple suivant : on prend un signal $x \in \mathbb{R}^N$ nul presque partout sauf en k endroits au plus, avec $k << N$. On dit de ce signal qu'il est *parcimonieux*. S'il n'y a pas de connaissance a priori de la position de ces k endroits, il est nécessaire de prendre les N échantillons du signal ; mais a posteriori, l'information peut se réduire à $2k$ éléments : l'instant et l'amplitude de ces k éléments non nuls. Les données collectées sont a posteriori plus nombreuses que nécessaire pour coder l'information réelle contenue dans le signal, particulièrement si on sait à l'avance qu'il n'y aura pas plus de k éléments non nuls. Dans ce cas où les zéros sont directement observés, il serait bien sûr possible de détecter les éléments non nuls à l'aide d'un détecteur de seuil et de ne transmettre que l'information essentielle, mais dans les exemples moins simples, ce n'est plus possible.

Imaginons maintenant prendre la projection de ce signal x sur une matrice $\Phi \in \mathbb{R}^{m \times N}$, avec $m \leq N$, et dont on ne conserverait que l'information $y = \Phi x$. Par exemple, une matrice dont les éléments sont issus d'un tirage aléatoire gaussien dont on s'est assuré qu'elle était de rang m (ce qui est très probable). Dans le cas particulier où $m = N$, il est facile de retrouver x à partir de y en utilisant l'inverse de la matrice Φ. Mais lorsque $m < N$, l'inverse de Φ n'existe plus. Une solution souvent rencontrée pour ce genre de problème est d'utiliser l'inverse généralisée de Moore-Penrose, notée Φ^+, qui minimise la

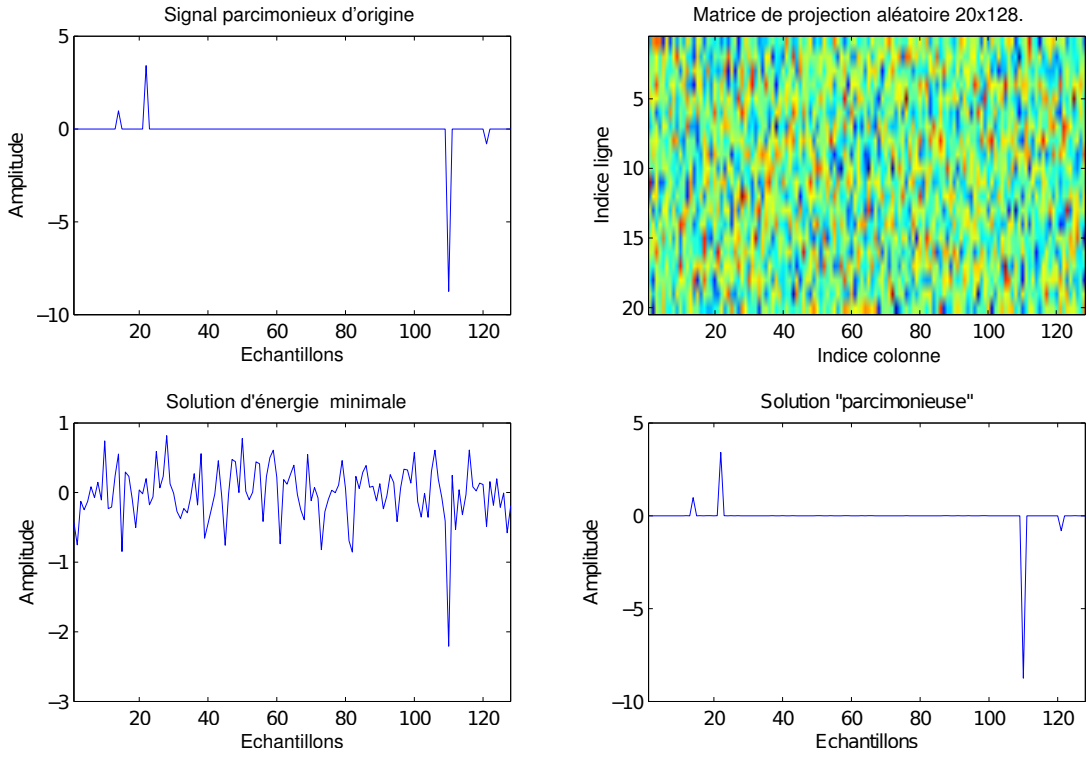

FIGURE 1.2 – Exemple d'un signal de longueur $N = 128$ contenant $k = 4$ valeurs non nulles en (a). La figure (b) représente une matrice aléatoire de taille $m \times N$ avec $m = 20$. En (c), la solution issue de la pseudo-inverse de Moore-Penrose et en (d) le résultat de l'algorithme Matching Pursuit appliqué à y et Φ.

norme ℓ_2 (c'est-à-dire l'énergie) de la solution. Dans le cas où le rang de la matrice est m, cette solution s'écrit de la manière suivante :

$$\hat{x} = \Phi^{+} y = \Phi^{T} \left(\Phi\Phi^{T}\right)^{-1} y, \tag{1.1}$$

mais ce résultat n'est pas identique à l'original : il n'est plus parcimonieux comme l'illustre la figure 1.2 (c).

Pourtant, de manière intuitive, si $m \geq 2k$, il y a potentiellement suffisamment d'information contenue dans y pour retrouver x. De plus, la probabilité que la projection de x sur une ligne de la matrice aléatoire soit nulle est faible. Chacune de ces projections va donc comporter une signature non nulle du signal, alors que si on projetait sur la matrice identité de taille N, amputée de $N - m$ lignes, ce qui revient à ne prendre que m échantillons sur les N nécessaires, il y aurait peu de chance que la projection porte une information autre que la présence d'un élément nul à un endroit donné. [1]

Les méthodes proposées dans la littérature (cf. section 2.2) sur le codage parcimonieux permettent de résoudre ce problème, et de reconstruire le signal comme illustré par la fi-

1. Une projection a $\frac{k}{N}$ chances de voir un signal, soit $1 - \frac{k}{N}$ chance de ne rien voir. La probabilité que la projection sur la matrice identité amputée de $N - m$ lignes soit nulle est donc de $(1 - \frac{k}{N})^m$.

gure 1.2 (d). On peut donc reconstruire un signal en ayant réduit le nombre d'échantillons (au sens large, ce ne sont plus des échantillons temporels) mais au prix d'une hypothèse : le signal est parcimonieux relativement à une base que l'on suppose connue. Dans notre exemple, le signal est parcimonieux dans sa représentation directe (sur la base canonique), mais ce n'est pas le cas des images compressées par la méthode JPEG, ou des musiques compressées par MP3. Dans ces cas là, on a choisi a priori des méthodes de représentations (transformée de Fourier discrète, transformée en cosinus discret) qui permettent une concentration de l'information sur quelques coefficients. On peut évidemment faire de même avec le codage parcimonieux, mais s'il n'existe pas de tels a priori, peut-on identifier un dictionnaire adéquat ?

Outre cette contrainte d'existence et de connaissance d'une méthode de représentation parcimonieuse du signal étudié, il y a un second obstacle à l'application du codage parcimonieux : les méthodes de reconstruction ne sont pas évidentes et coûtent cher en calcul. Puisque l'on conserve l'information du signal dans sa forme compressée, se poser la question suivante est alors une évidence : peut-on exploiter l'information contenue dans le signal compressé sans passer par l'étape de reconstruction ?

1.2 Organisation du document

Dans une première partie du document, nous nous appuyons sur l'état de l'art pour formuler les concepts du codage parcimonieux et introduire les problèmes que nous allons aborder par la suite.

- Dans le chapitre 2, nous présentons les notions relatives au codage parcimonieux, en ayant une approche intuitive. À l'aide d'exemples simples, nous présentons les possibilités de compression du signal et de reconstruction et introduisons empiriquement quelques propriétés que doivent satisfaire le signal et la matrice d'observation pour que le système fonctionne. Nous présentons aussi quelques méthodes de reconstruction.
- Après cette approche empirique, le chapitre 3 revient sur les justifications théoriques énoncées dans la littérature. Cela permet de formaliser les propriétés nécessaires pour que la reconstruction soit exacte, ou lorsqu'elle ne l'est pas, de quantifier l'erreur.

Après cette partie de présentation du codage parcimonieux, nous nous intéressons à la mise en œuvre et aux problèmes soulevés.

- Les simulations sur des exemples synthétiques simples du chapitre 4 permettent d'avoir une idée des ordres de grandeur du nombre de composantes du signal et du nombre d'observations nécessaires au bon fonctionnement du codage parcimonieux. Les algorithmes présentés dans les chapitres précédents sont comparés et la possibilité d'utiliser une matrice binaire validée. Enfin, un dernier exemple sur des signaux enregistrés permet de se rendre compte que dans un cas moins idéal, les résultats ne sont pas aussi bons mais néanmoins acceptables.

- Le chapitre 5 aborde le problème de la réalisation d'un capteur intégrant une technologie de codage parcimonieux. Nous utilisons une approche numérique, en faisant le choix de travailler après un échantillonnage dans les règles du signal, en joignant un microcontrôleur au capteur.
- Le chapitre 6 revient sur le problème de la définition du dictionnaire de représentation parcimonieuse par apprentissage. Les chapitres précédents ont mentionné qu'un tel dictionnaire est indispensable au fonctionnement du codage parcimonieux. On s'intéresse ici à l'apprentissage de dictionnaire à partir d'une banque de signaux représentative du système qui doit être codé.
- Enfin, les discussions sur les méthodes de reconstruction ont mis en avant le fait que celles-ci étaient coûteuses. Le chapitre 7 traite d'une possibilité d'exploitation du signal compressé en évitant cette étape, au travers d'un exemple de classification de signaux neuronaux. On étudie ensuite la conservation de la norme par projection, qui explique le bon fonctionnement de l'exemple.

Chapitre 2

Parcimonie et codage parcimonieux

Résumé : *Dans ce chapitre, on introduit les principes du codage parcimonieux, en s'appuyant sur l'état de l'art existant. On détaille la notion de parcimonie de la représentation du signal, ainsi que le fonctionnement du codage et du décodage parcimonieux.*

On a choisi de s'intéresser au codage parcimonieux dans la perspective d'avoir une méthode de compression efficace et peu coûteuse, en étant prêt à faire deux concessions : les signaux considérés doivent pouvoir être représentés de manière parcimonieuse (ce qui impose de connaître ou d'apprendre le dictionnaire) et la reconstruction (décompression) peut être difficile. Dans ce chapitre, on commence par définir la notion de parcimonie, inhérente au principe de codage parcimonieux. Ensuite, on détaille, à l'aide d'exemples, le fonctionnement du codage parcimonieux, en séparant deux notions : le codage et le décodage. L'objectif de ce chapitre est de reformuler l'état de l'art pour introduire le codage parcimonieux. S'il y a peu de références mentionnées, celles-ci sont citées dans la suite du document.

2.1 Qu'est ce que la parcimonie ?

La *parcimonie* est l'hypothèse essentielle sur laquelle repose le codage parcimonieux. De manière empirique, on parle de signal parcimonieux lorsqu'il est possible de décrire le signal à partir d'un faible nombre de composantes élémentaires connues a priori. Dans cette section, nous établissons les méthodes de représentation et de quantification permettant d'identifier un signal comme parcimonieux, c'est-à-dire que l'on peut décomposer de manière parcimonieuse dans un espace de représentation donné.

2.1.1 Bases et dictionnaires

Considérons un signal $x \in \mathbb{R}^N$. Classiquement, ce signal est représenté dans la base canonique de $\mathbb{R}^N$, dont la représentation matricielle est la matrice identité de taille N. Le vecteur x est décrit comme $x = \sum_{i=1}^{N} x_i e_i$, où $e_i = (\delta_{1,i}, \delta_{2,i}, ..., \delta_{N,i})^T$, où $\delta_{i,j}$ est le *symbole de Kronecker* : $\delta_{i,j} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$. La base canonique n'est pas la seule base de représentation possible : toute autre base Ψ de $\mathbb{R}^N$ peut décrire x, sous la forme

$$x = \Psi\alpha = \sum_{i=1}^{N} \alpha_i \Psi_i,$$

où les Ψ_i sont les colonnes de Ψ. Le changement de base peut faciliter l'interprétation des signaux : par exemple, la transformée de Fourier permet une analyse harmonique du signal, et une fréquence pure est représentée par seulement deux coefficients complexes : l'énergie est concentrée sur quelques éléments, plutôt que répartie sur l'ensemble. On dit que la représentation de x est parcimonieuse si $k = \text{Card}\{\alpha_i \neq 0 \,|\, i \in (1, N)\} << N$.

Toutefois, il n'est pas toujours aisé de trouver une base dans laquelle les signaux possèdent naturellement une représentation parcimonieuse. On peut alors ne plus se restreindre à une base et ajouter de la redondance en augmentant le nombre de colonnes de la matrice Ψ. Dans ce cas là, on parle de repère (ou *frame* :)

Définition 4 (Repère de $\mathbb{R}^N$) *Une famille $\Psi_i, i \in I$, où I est un ensemble d'indices, est un repère de $\mathbb{R}^N$ s'il existe A, B, avec $0 < A < B < +\infty$ tels que pour tout $x \in \mathbb{R}^N$:*

$$A\|x\|_2^2 \leq \sum_{i \in I} | < x, \Psi_i > |^2 \leq B\|x\|_2^2.$$

Si $A = B$, on dit que le repère est ajusté (tight frame.)

Dans le cas où Ψ est sur-complet, c'est-à-dire que le nombre de colonnes est supérieur à la longueur de celles-ci, et où les Ψ_i sont normés, on parle du *dictionnaire* Ψ et les colonnes Ψ_i sont appelées les *atomes* du dictionnaire. Dans ce cas là, la décomposition de x n'est plus unique.

Remarque : il est évident que sont considérés ici des ensembles de signaux issus d'un même phénomène que l'on cherche à décrire de manière parcimonieuse. Si on ne considérait qu'un seul signal, donné et fixe, une base de décomposition triviale serait celle construite à partir de l'assemblage du vecteur lui-même et de $N - 1$ vecteurs orthogonaux entre eux (comme les éléments de base e_i de la base canonique).

2.1.2 Comment quantifier la parcimonie ?

2.1.2.1 Parcimonie et normes

S'il est simple de décrire l'idée de la parcimonie, il est moins évident de quantifier, par une méthode de mesure, la parcimonie d'un signal. Une mesure intuitive est celle du nombre d'éléments non nuls du signal, que l'on peut noter comme suit :

Définition 5 (Norme ℓ_0) *Soit un vecteur $x \in \mathbb{R}^N$, sa norme ℓ_0 se définit par :*

$$\|x\|_0 = \text{Card}\{x_i \neq 0 \,|\, i \in (1, N)\}$$

Cette mesure est appelée abusivement *norme* ℓ_0 bien qu'il ne s'agisse pas d'une norme puisque la propriété d'homogénéité n'est pas vérifiée. [1]

Si un vecteur x est tel que $\|x\|_0 = k$ avec $k << N$, on dit que x est k-parcimonieux (k-*sparse* :) il y a k éléments non nuls dans le vecteur x.

Cependant, si cette norme ℓ_0 semble adaptée à la quantification de la parcimonie d'un signal, le fait que tout élément non nul ait la même importance dans la valeur de la norme peut se révéler un obstacle dans la pratique : la présence de bruit, une approximation due à un effet d'arrondi, peut remettre en cause la valeur et mener à une situation où $\|x\|_0 \simeq N$ alors que l'énergie du signal reste concentrée sur un faible nombre de composantes.

Pour pallier ce défaut, il est courant de recourir à d'autres mesures moins évidentes :

Définition 6 (Norme ℓ_p) *Soit un vecteur $x \in \mathbb{R}^N$, et $p > 0$, sa norme ℓ_p se définit par :*

$$\|x\|_p = \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}}$$

Remarque : il ne s'agit en vérité d'une norme que si $p \geq 1$. Si $0 < p < 1$, l'inégalité triangulaire n'est pas vérifiée [2]. On peut appeler ceci une quasi-norme, mais par abus de langage, on utilise dans la suite de ce document le terme norme quelle que soit la valeur de p.

Choisir la valeur de p permet de régler l'importance donnée aux petits coefficients. En effet, si on appelle f la fonction associant à $x \in \mathbb{R}^N$ sa norme ℓ_p : $f(x) = \|x\|_p$, une approximation de $f(x)$ est donnée par

$$f(x + d_x) \approx f(x) + \nabla f(x) \cdot d_x.$$

1. En effet, $\forall \lambda \in \mathbb{R}^* \backslash \{1\}, \forall x \in \mathbb{R}^N, x \neq 0, \|\lambda x\|_0 \neq |\lambda| \|x\|_0$.
2. Cependant, la norme ℓ_p élevée à la puissance p, $\|x - y\|_p^p$, vérifie l'inégalité triangulaire et est donc une distance.

où $\nabla f(x)$ est le gradient de f en x :

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1} \frac{\partial f(x)}{\partial x_2} ... \frac{\partial f(x)}{\partial x_N} \right].$$

Or,

$$\frac{\partial f(x)}{\partial |x_i|} = \left(\frac{|x_i|}{\|x\|_p} \right)^{p-1}.$$

Ceci met en avant le fait qu'une petite composante n'a pas le même effet sur la norme du vecteur selon la valeur de p : en effet, lorsque $p < 1$, la moindre variation d'une composante de faible amplitude entraine une forte croissance de la norme (d'autant plus forte que p est proche de 0), alors que lorsque $p > 1$, l'augmentation de la norme est minime. Cette norme ℓ_p peut donc être utilisée, avec $0 < p < 1$, pour distinguer deux vecteurs selon la parcimonie de leur représentation.

2.1.2.2 Illustration

La figure 2.1 permet de visualiser l'effet du choix de p en représentant l'évolution, en fonction de ce paramètre p, de la norme ℓ_p des quatre vecteurs suivants :

$$a = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} ; \; b = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix} ; \; c = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} \text{ et } d = \begin{bmatrix} 0.95 \\ 0.05 \\ 0 \\ 0 \end{bmatrix}.$$

a, b et c sont tels que $\|a\|_1 = \|b\|_1 = \|c\|_1$: cela permet de mettre en avant le changement autour de $p = 1$, où les lignes se croisent et l'ordre s'inverse. On remarque que lorsque p est proche de 0, l'ordre établi par la norme ℓ_0 se retrouve et le nombre de coefficients non nuls permet de distinguer les vecteurs. Cependant, il y a une légère différence entre $\|b\|_p$ et $\|d\|_p$ alors que $\|b\|_0 = \|d\|_0$: la norme ℓ_p permet de rajouter une finesse de distinction en prenant en compte la répartition de l'énergie entre les éléments non nuls. De plus, même si $\|d\|_p \geq \|b\|_p$ pour $p \geq 1$, on arrive à dire que d est plus parcimonieux que b si on choisit une norme ℓ_p avec p suffisamment proche de 0.

Cet exemple illustre le fait que la norme ℓ_p peut être utilisée comme mesure de parcimonie et que choisir une valeur de p proche de 0 permet de se rapprocher du comportement de la mesure intuitive qu'est la norme ℓ_0 tout en ajoutant une nuance selon la valeur des éléments non nuls.

2.1.2.3 Parcimonie et entropie

Les coefficients de représentation peuvent aussi s'envisager comme la probabilité d'utilisation d'un atome donné pour la décrire le signal. Pour un vecteur $x \in \mathbb{R}^N$ que l'on

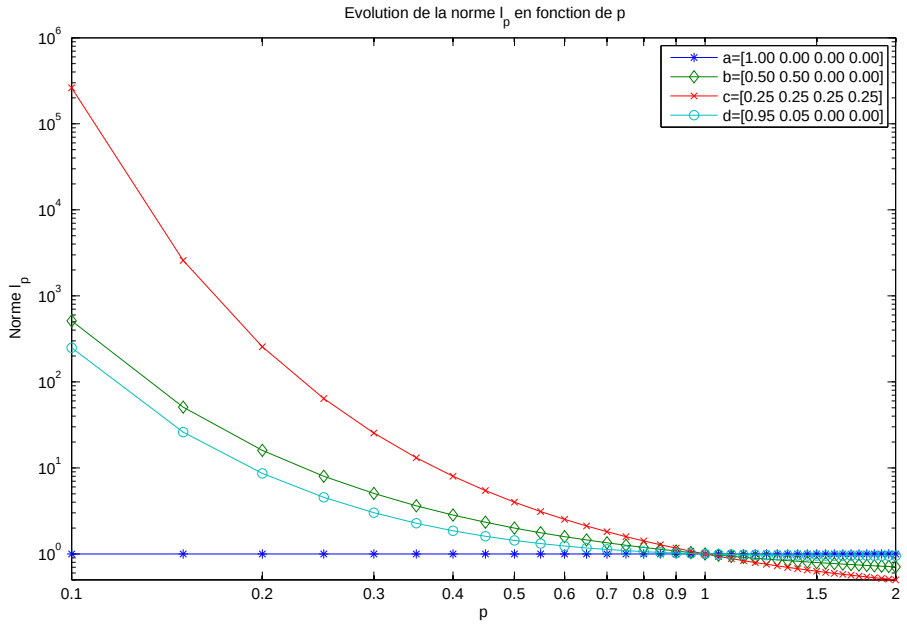

FIGURE 2.1 – Évolution de la norme ℓ_p, pour p variant de 0.1 à 2, des 4 vecteurs suivants : $a = [1\,0\,0\,0]$; $b = [0.5\,0.5\,0\,0]$; $c = [0.25\,0.25\,0.25\,0.25]$; $d = [0.95\,0.05\,0\,0]$.

normalise en norme ℓ_1, chaque élément $|x_i|$ peut s'interpréter comme une probabilité d'utiliser cet i-ème atome de la base $\mathbb{R}^N$, avec $\sum_{i=1}^{N} |x_i| = 1$. L'entropie permet de caractériser la distribution d'une loi de probabilité : plus elle est grande, plus il y a de désordre. Si le support de la loi de probabilité est borné, la distribution qui maximise l'entropie est la distribution uniforme. Inversement, si l'entropie est faible, il y a peu de désordre : la distribution est concentrée sur quelques états. L'entropie de Rényi est une mesure d'entropie :

Définition 7 (Entropie de Rényi) *Soit une variable aléatoire discrète X à N états de probabilités respectives p_1, p_2, ..., p_N, l'entropie de Rényi H_α de X pour $\alpha > 0$ est définie comme :*

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{N} p_i^\alpha \right)$$

Si on s'intéresse au logarithme de la norme ℓ_p :

$$\log \|x\|_p = \log \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}} = \frac{1}{p} \log \left(\sum_{i=1}^{N} |x_i|^p \right),$$

on retrouve l'expression de l'entropie de Rényi H_p, au facteur multiplicatif $\frac{p}{1-p}$ près. Ce facteur est strictement positif si $0 < p < 1$. De fait, minimiser la norme ℓ_p est analogue au fait de minimiser l'entropie de Rényi de la répartition de l'amplitude des coefficients sur les atomes du dictionnaire. Cela donne une autre explication au fait que la norme ℓ_p avec

$0 < p < 1$ constitue un indicateur de parcimonie.

2.2 Codage parcimonieux et décodage

Le codage parcimonieux, comme un système de compression classique, nécessite un décodage qui lui correspond (fig. 2.2). Le codage est une opération de projection sur la matrice d'observation Φ. Cette matrice est nécessaire à l'étape de décodage pour traiter l'observation $y = \Phi x$ et retrouver x. En outre, il faut disposer du dictionnaire Ψ de décomposition parcimonieuse du signal x. La partie suivante explique pourquoi ce dictionnaire, même s'il n'est pas utilisé par l'opération de codage, conditionne le choix de la matrice d'observation Ψ. Ensuite, le principe de reconstruction du signal est détaillé.

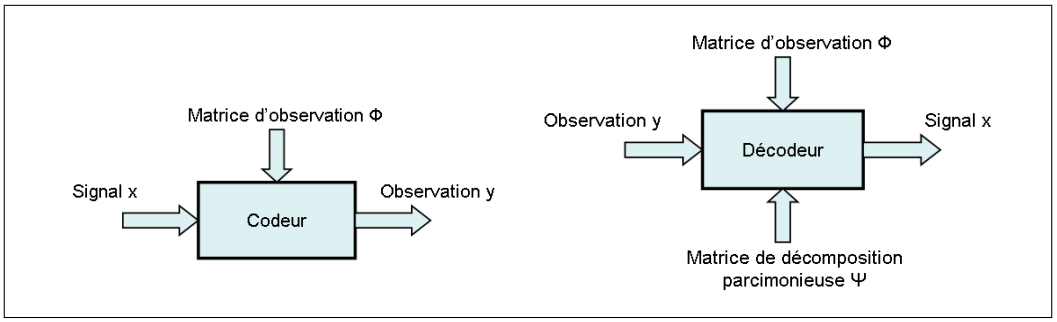

FIGURE 2.2 – Schéma de principe du codage parcimonieux, représenté par un codeur et un décodeur.

2.2.1 Codage : les diverses matrices d'observation

L'exemple présenté dans l'introduction (fig. 1.2 p. 12) a illustré, sur un exemple particulier, la possibilité de reconstruire un signal parcimonieux à partir d'un nombre réduit de projections sur une matrice aléatoire[3] (Φ, de dimension $m \times N$, $m < N$), ce qui a considérablement réduit le nombre d'échantillons utilisés (de 128 à 20). Pourquoi avoir fait le choix d'une matrice aléatoire ? Pourquoi est-ce que cela fonctionne ?

Supposons que le signal peut se représenter de manière parcimonieuse dans un dictionnaire Ψ. Pour que le codage soit bon, il faut que la matrice d'observation Φ "explore" toutes les colonnes du dictionnaire Ψ, ce que permet généralement une matrice aléatoire. Dans notre exemple, le dictionnaire est la matrice identité ($\Psi = Id_N$) : une séquence aléatoire gaussienne, centrée, ne peut pas être représentée de manière parcimonieuse dans la base canonique. En effet, la probabilité qu'un élément de la séquence soit nul vaut 0. On explore donc bien tout le dictionnaire avec une telle matrice. A contrario, lorsqu'on choisit m

3. La matrice est déterminée à partir d'un tirage aléatoire à la suite duquel elle est fixée définitivement. Dans ce document, l'expression *matrice aléatoire* désigne systématiquement ce type de matrice.

échantillons temporels, la matrice de projection est la matrice identité amputée de $N-m$ lignes. Chacune des lignes restantes correspond à une ligne la base canonique. Les $N-m$ autres lignes de cette base canonique ne sont pas observées.

Pour résumer, il faut que les éléments de la matrice d'observation ne puissent pas être représentés parcimonieusement sur le dictionnaire de décomposition parcimonieuse du signal, et inversement. Ceci peut se mesurer à l'aide de la cohérence mutuelle, définie comme ceci :

Définition 8 (Cohérence mutuelle) *Soient deux matrices Φ de taille $m \times N$ et Ψ de taille $N \times L$, la cohérence mutuelle μ entre ces deux matrices est définie comme le maximum de corrélation entre les lignes Φ^i de Φ et les colonnes Ψ_j de Ψ :*

$$\mu(\Phi, \Psi) = \max_{\substack{1 \leq i \leq m \\ 1 \leq j \leq L}} \frac{| < \Phi^i, \Psi_j > |}{\|\Phi^i\|_2 \|\Psi_j\|_2} ;$$

Dans le cas où une des deux matrices est une base orthonormée, on a le résultat suivant : $\mu \geq \frac{1}{\sqrt{N}}$ [GN03]. En effet, si Φ est une base orthonormée et Ψ_j un vecteur quelconque, alors $\sum_{k=1}^{N} | < \Phi^i, \Psi_j > |^2 = \|\Psi_j\|_2^2$. Par conséquent, $\max_{1 \leq k \leq N} | < \Phi^i, \Psi_j > |^2 \geq \frac{\|\Psi_j\|_2^2}{N}$. D'où le résultat. Dans le pire cas, $\mu = 1$.

Si les coefficients d'une des matrices sont aléatoires, indépendants, distribués suivant une loi normale centrée, et l'autre matrice une base orthogonale quelconque, le calcul de l'espérance de μ n'est pas immédiat. Cependant, de simples simulations numériques montrent que si on n'atteint pas la valeur limite $1/\sqrt{N}$, on en reste assez proche ($\mathbb{E}\{\mu\} \approx \frac{c}{\sqrt{N}}$ avec $c < 4$ pour $N < 2000$, c croît avec N). [4] Dans le cas inapproprié où l'on choisit m échantillons pris au hasard dans le temps, la matrice Φ est alors représentée par une sélection de m lignes de la matrice identité : la cohérence avec la matrice identité vaut alors 1 : les deux matrices sont cohérentes et c'est le pire cas possible. Cela donne une autre justification de l'inadaptation d'un tel choix.

En poursuivant cette idée d'incohérence entre les matrices d'observation et de représentation, il y a une matrice qui est particulièrement bien adaptée à l'observation d'un signal impulsionnel : la matrice de la transformée de Fourier discrète [5], dont on aura choisi m lignes (aléatoirement). En effet, la cohérence mutuelle entre cette matrice de Fourier discrète et la base canonique, en dimension N, vaut $\mu = \frac{1}{\sqrt{N}}$, c'est-à-dire le minimum possible lorsque l'on a au moins une base orthonormée [GN03]. Ceci est illustré par la figure 2.3, qui montre un résultat tout aussi efficace qu'avec une matrice aléatoire.

4. *Un approfondissement en annexe est à venir.*

5. Matrice $F = (f_{m,n})$ de taille $N \times N$, où les éléments valent

$$f_{m,n} = \frac{e^{-2\iota\pi \frac{(m-1)(n-1)}{N}}}{\sqrt{N}}, \text{ où } \iota^2 = -1 \text{ et } m = 0, ..., N-1 \,; n = 0, ..., N-1.$$

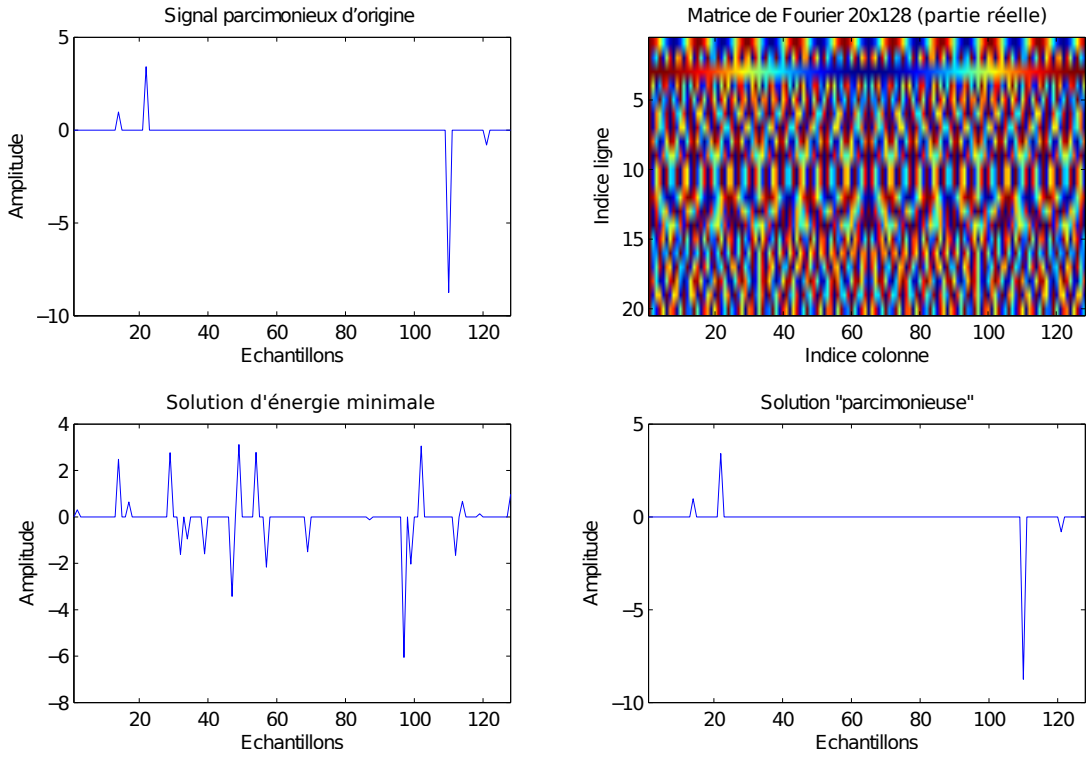

FIGURE 2.3 – Exemple d'un signal x de longueur $N = 128$ contenant $k = 4$ valeurs non nulles en (a). Ce signal est codé par y, résultat de la projection de x sur Φ, une matrice de TFD incomplète de taille $m \times N$ avec $m = 20$, représentée en (b). En (c), on peut voir la solution obtenue à l'aide de la pseudo-inverse de Moore-Penrose à partir de y et de Φ, et en (d) celle obtenue via l'algorithme Orthognal Matching Pursuit (cf. p. 26).

Nous avons, jusqu'ici, considéré des signaux parcimonieux dans la base naturelle ($\Psi = \mathbb{1}$). En général, les signaux parcimonieux le sont dans une autre base (ou dictionnaire) ($\Psi \neq \mathbb{1}$) : il existe alors α tel que $\|\alpha\|_0 << N$ et $x = \Psi\alpha$. Par exemple, le signal x peut être parcimonieux dans l'espace de Fourier discret, c'est-à-dire être la somme de k sinusoïdes [6], comme l'illustre la figure 2.4. Dans ce cas-là, on ne recherche plus le signal x mais le vecteur α parcimonieux, en supposant que l'observation y est obtenue à partir de $\Phi\Psi\alpha$. Il est tout à fait possible ici d'utiliser comme matrice de projection la matrice identité amputée de $N - m$ lignes choisies au hasard – ceci correspond donc à choisir au hasard m échantillons temporels – puisque la cohérence entre cette matrice et la matrice de Fourier discrète est minimale : le produit scalaire entre le vecteur représentant le choix du i-ème échantillon temporel avec la j-ème colonne de la matrice de Fourier vaut en effet $|e^{-2\imath\pi\frac{ij}{N}}/\sqrt{N}| = 1/\sqrt{N}$. Le rôle des deux matrices est inversé par rapport à l'exemple précédent, mais le résultat est identique : on peut retrouver et reconstruire la somme d'un petit nombre de sinusoïdes à partir de quelques échantillons temporels. On remarque qu'il aurait été tout à fait possible d'utiliser ici aussi une matrice aléatoire, pour arriver à un

6. Pour cet exemple, il faut remarquer que la fréquence de la sinusoïde ne doit pas être quelconque, sinon sa représentation dans l'espace de Fourier discret ne sera pas parcimonieuse, le pire cas étant lorsque la fréquence est égale à $f = (k + 0.5)\frac{F_e}{N}$, $k \in \mathbb{Z}$. On suppose donc que $f = k\frac{F_e}{N}$, $k \in \mathbb{Z}$

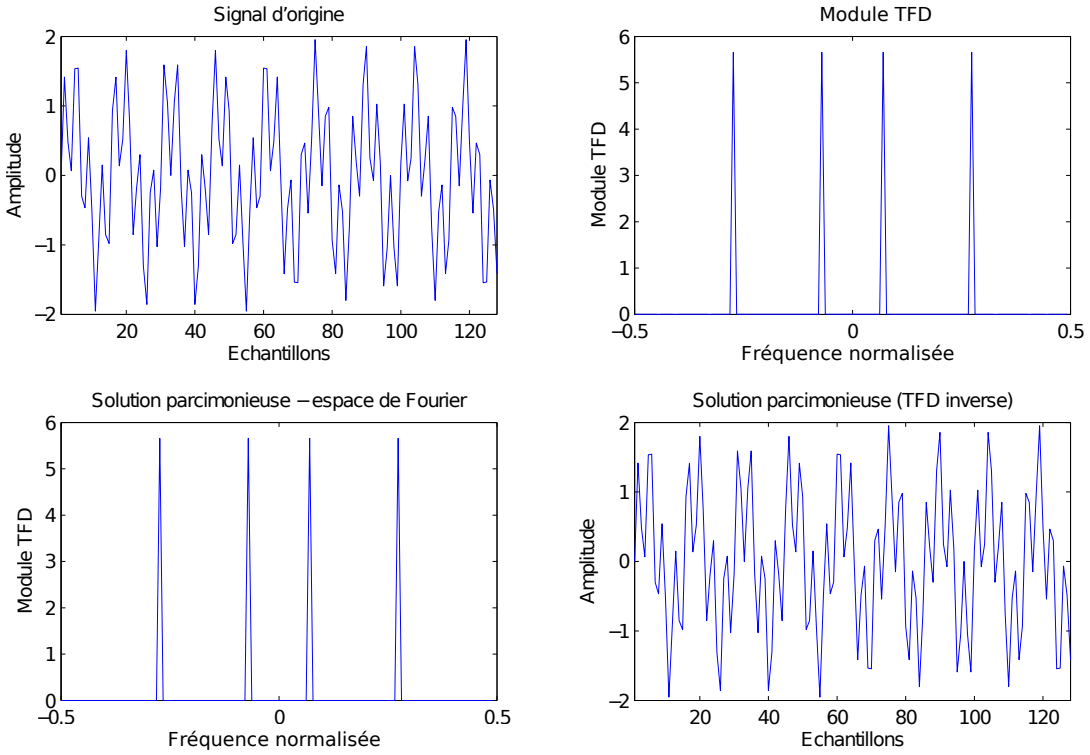

FIGURE 2.4 – Exemple d'un signal x de longueur $N = 128$, somme de 2 sinusoïdes, en (a) dont le module de la transformée de Fourier discrète (le vecteur α) est représenté en (b). Ce signal est codé par le choix de $m = 20$ échantillons au hasard, représentés par m lignes de la matrice identité. À partir de cette observation, on peut reconstruire α (module affiché en (c)) par l'algorithme OMP et donc retrouver le signal original x par transformée de Fourier inverse (d).

résultat identique. En revanche, il n'aurait pas été possible d'utiliser une matrice de TFD réduite comme dans l'exemple précédent, puisque celle-ci correspond à Ψ.

Au final, la matrice de codage Φ dépend du dictionnaire de représentation parcimonieuse du signal Ψ, au sens où on doit satisfaire le critère d'incohérence entre les deux. Un choix de matrice d'observation correct pour un type de signal peut se révéler totalement inadapté pour un autre signal, comme l'ont illustré les précédents exemples. Des méthodes ont été proposées [Ela07] afin d'optimiser le choix de la matrice en fonction du dictionnaire en visant à minimiser la cohérence entre les deux matrices. Celles-ci semblent efficaces, même si ce critère n'est pas celui qui apporte les meilleures certitudes sur la performance du codage parcimonieux (cf. p. 38). Ceci étant, choisir une matrice issue d'un tirage aléatoire permet généralement d'obtenir un codeur efficace quel que soit le dictionnaire Ψ envisagé : cela permet de concevoir un codeur "universel" [CT06].

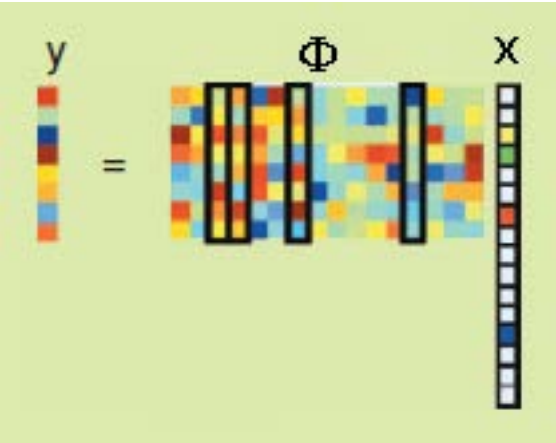

FIGURE 2.5 – Exemple de projection d'un vecteur k-parcimonieux sur une matrice quelconque. Cela sélectionne les k colonnes correspondantes de la matrice. Illustration R. Baraniuk [Bar07]

2.2.2 Décodage : les algorithmes de reconstruction

Les exemples précédents ont illustré le fait qu'il est possible, sous certaines conditions, de reconstruire un signal parcimonieux x, à partir de l'observation

$$y = \Phi x.$$

La matrice Φ est de dimension $m \times N$ avec $m < N$: elle est donc de rang inférieur à N et n'est pas inversible. Cela implique qu'il existe une infinité de solutions $\hat{x}$ telles que $\Phi\hat{x} = \Phi x$, mais $\hat{x} \neq x$. Le problème du décodage est de déterminer laquelle parmi ces solutions correspond à x. La pseudo-inverse de Moore-Penrose est une méthode classique qui permet d'obtenir la solution de norme ℓ_2 minimale :

$$\hat{x} = \arg\min_x \|x\|_2 \text{ tel que } y = \Phi x.$$

Le résultat est donné par l'équation 1.1, rappelée ci-dessous, à condition que la matrice Φ soit de rang m :

$$\hat{x} = \Phi^+ y = \Phi^T \left(\Phi\Phi^T\right)^{-1} y, \tag{2.1}$$

Les exemples (fig. 1.2 et fig. 2.3) ont illustré le fait que cette méthode est inadaptée à la recherche d'un signal parcimonieux. En effet, la minimisation de la norme ℓ_2 n'invite pas à la parcimonie de la solution puisqu'une telle norme ne pénalise pas les petits coefficients et qu'au contraire, la solution a tendance à répartir l'énergie sur tout le signal : l'hypothèse de l'existence d'une décomposition parcimonieuse du signal n'est pas prise en compte.

Pour exploiter cette information a priori, il faut donc utiliser une approche qui pénalise les solutions non parcimonieuses, à l'aide des méthodes de mesure présentées en 2.1.2. L'approche naturelle est donc de chercher à minimiser la norme ℓ_0 de la solution, ce qui se

traduit par le problème suivant :

$$(P_0) : \qquad \hat{x} = \arg\min_x \|x\|_0 \text{ tel que } y = \Phi x \tag{2.2}$$

Toutefois, la recherche d'une telle solution ne peut se faire en temps polynomial car c'est un problème NP-complet [Nat95]. La recherche d'une solution k-parcimonieuse nécessite d'énumérer toutes les possibilités de k éléments non nuls : il y en a $\binom{N}{k}$. D'autres approches sont envisageables, notamment en utilisant les autres mesures de parcimonie précédemment citées, et certaines d'entre elles sont décrites dans la suite de ce chapitre.

2.2.2.1 Les algorithmes gloutons

Un algorithme glouton est un algorithme itératif qui effectue à chaque étape le choix localement optimal. Ceci n'implique pas que le résultat final soit optimal, notamment parce que l'algorithme peut faire un mauvais choix au départ qui l'empêchera de converger par la suite. Par exemple, un algorithme glouton peut être utilisé pour rechercher la manière optimale (au sens du plus petit nombre de pièces) de rendre la monnaie, en supposant qu'il y ait suffisamment de pièces, en choisissant à chaque fois la pièce ayant la plus grande valeur inférieure au montant dû restant.

Le codage parcimonieux est présenté comme un produit matriciel sous la forme $y = \Phi x$. Cependant, on peut le voir d'une manière différente, comme illustré sur la figure 2.5 : l'observation y d'un signal x k-parcimonieux est la somme de k colonnes de la matrice Φ, pondérée par les valeurs non nulles de x correspondantes, on peut l'écrire :

$$y = \sum_{i=1}^{N} x_i \phi_i = \sum_{i|x_i \neq 0} x_i \phi_i.$$

Le problème de décodage revient donc à déterminer quelle combinaison de colonnes constitue l'observation et ceci en utilisant le moins de colonnes possible puisqu'on fait l'hypothèse que le signal est parcimonieux, ce qui est similaire à la recherche des pièces nécessaires pour rendre la monnaie : on cherche les pièces dont la valeur se rapproche le plus du montant dû.

Ce problème de la recherche d'une solution parcimonieuse peut donc être traité par un algorithme glouton : Mallat et Zheng ont proposé l'algorithme Matching Pursuit [MZ93]. Matching Pursuit procède de la manière suivante (cf. annexe A) : à partir d'un résidu, qui a comme valeur initiale l'observation compressée, l'algorithme sélectionne l'atome du dictionnaire qui se rapproche le plus de l'observation, puis retire du résidu la contribution de cet atome ; et cela est répété jusqu'à ce que le résidu soit suffisamment petit relativement à l'observation de départ. La solution est, par construction, parcimonieuse puisque après n itérations, le signal obtenu a au plus n composantes non nulles.

L'exemple de la figure 2.6 illustre l'évolution de la reconstruction d'un signal tel que $\|x\|_0 = 5$. On constate que les 5 premières itérations sélectionnent bien 5 atomes différents. Ensuite, aucun nouvel atome n'est sélectionné, les itérations suivantes permettent d'ajuster les amplitudes par adjonction d'un atome déjà utilisé, avec un coefficient faible, jusqu'à converger vers l'amplitude exacte. Cependant, si MP fonctionne parfaitement dans le cas où Φ est une base orthogonale, Chen *et al* [CDS01] citent plusieurs exemples pour lesquels MP ne fonctionne pas car les premiers choix sont erronés, et les itérations suivantes ne cherchent qu'à compenser cette erreur. On retient l'exemple de la somme de deux sinusoïdes de fréquences proches lorsque le dictionnaire est surcomplet, ou bien l'exemple créé de toutes pièces par [DT96], où on adjoint à la base canonique une colonne composée d'une combinaison linéaire de toutes les autres. Pour reprendre l'exemple du rendu de monnaie, une situation où la méthode gloutonne qui consiste à rendre la pièce de plus forte valeur n'est pas optimale est celui où le système de pièces n'est pas adapté : s'il n'y a que des pièces de 1, 3 et 4, pour rendre 6, l'algorithme rendrait 4, 1 et 1, alors que deux pièces de 3 suffisent.

Orthogonal Matching Pursuit [PRK93] est une évolution de Matching Pursuit. Le problème de Matching Pursuit est qu'il est possible de sélectionner plusieurs fois le même atome. OMP permet de pallier ce défaut. Le début de l'itération est identique à MP : on commence par choisir l'atome du dictionnaire le plus corrélé avec le résidu. Ensuite, on assure que l'ensemble des atomes sélectionnés est orthogonal au résidu en recalculant la décomposition sur l'ensemble des atomes sélectionnés par l'algorithme. Concrètement, à la k-ème itération, si on appelle Θ^k l'ensemble des k atomes sélectionnés, alors le signal reconstruit vaut $x^k = \arg\min_x \|y - \Theta^k x\|_2$. Le résidu r^k vaut alors $r^k = y - \Theta^k x^k$.

Contrairement à Matching Pursuit, qui peut nécessiter un nombre indéterminé d'itérations pour obtenir une reconstruction exacte d'un signal formé de k composantes [7], l'algorithme OMP, s'il converge, le fait en k itérations, au prix d'une itération plus coûteuse puisque nécessitant une minimisation de moindres carrés. De plus, OMP fonctionne correctement sur les exemples qui posent problème à MP [CDS01], mais il est aussi possible de construire des exemples où il n'est pas optimal. Cependant, en pratique, les résultats sont bons (cf. chapitre 4.1.2, [TG07]).

La famille des algorithmes gloutons est large et il en existe d'autres plus évolués, permettant notamment la sélection de plusieurs atomes par itération (StOMP, ROMP), voire le rejet d'atomes déjà sélectionnés (CoSaMP). Les travaux de D. Needell [Nee09] en présentent une étude approfondie.

7. En k itérations, on a un signal constitué de au plus k composantes, mais l'expérience montre que cela ne suffit pas à reconstruire le signal : il faut réitérer un certain nombre de fois avant de converger, cf. fig. 2.6.

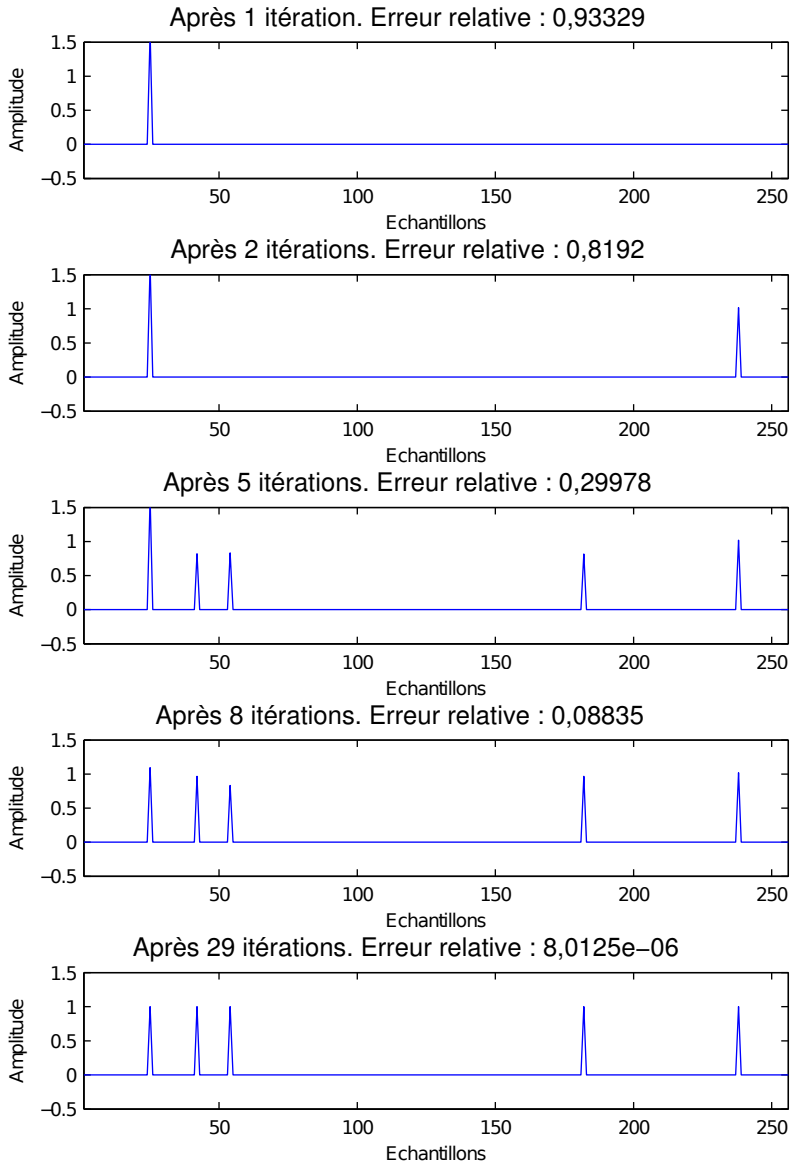

FIGURE 2.6 – Évolution de la reconstruction par l'algorithme Matching Pursuit d'un signal de longueur $N = 256$, nul partout sauf en 5 points, projeté sur une matrice aléatoire de taille $50 \times N$ (éléments issus d'un tirage aléatoire gaussien, i.i.d) dont les lignes ont été orthonormalisées.

2.2.2.2 Relaxation ℓ_1 et optimisation convexe

Historiquement, le codage parcimonieux s'est développé notamment lorsque Chen, Donoho et Saunders [CDS01] ont proposé, dans l'optique de rechercher des décompositions parcimonieuses dans un dictionnaire, la méthode suivante, appelée *Basis Pursuit* :

$$(P_1) : \qquad \hat{x} = \arg\min_x \|x\|_1 \text{ tel que } y = \Phi x \tag{2.3}$$

On pallie le problème de faisabilité de la recherche d'une solution au sens de la norme ℓ_0 en relâchant cette contrainte et en la remplaçant par la norme ℓ_1. Par la suite, les travaux

de Donoho [DH02] puis Candès *et al* [CRT06a, Can08] ont montré que, dans certains cas (avec une probabilité asymptotiquement égale à 1 pour les matrices aléatoires), résoudre (P_1) donne exactement la solution de (P_0) (éq. (2.2)) : ceci est discuté en section 3.1.1.

L'avantage de la formulation (P_1) est qu'elle peut se traduire par un problème d'optimisation convexe, si les coefficients sont réels [CR05], qui peut être résolu par programmation linéaire [CR05]. Une variante de *Basis Pursuit* appelée *Basis Pursuit DeNoising (BPDN)* traite le problème de la mesure bruitée, c'est à dire

$$y = \Phi x + w,$$

où $w \in \mathbb{R}^m$ est un terme de bruit inconnu tel que $\|w\|_2 \leq \varepsilon$. Le problème devient alors :

$$(BPDN) : \qquad \hat{x} = \arg\min_x \|x\|_1 \text{ tel que } \|y - \Phi x\|_2 \leq \varepsilon. \tag{2.4}$$

Celui-ci ne peut plus être traité comme un programme linéaire, mais comme un *SOCP (Second Order Cone Program)*. Une méthode souvent utilisée pour le résoudre est la méthode dite du point intérieur [CR05]. Le lecteur intéressé trouvera un descriptif détaillé de ces méthodes dans la documentation de la bibliothèque ℓ_1-MAGIC [CR05]. Cette bibliothèque fournit, entre autres, les méthodes l1eq_pd pour résoudre (P_1) et l1eq_qc pour résoudre $(BPDN)$.

2.2.2.3 Moindres carrés pondérés

Dans la section précédente, on a vu que la recherche d'une solution de norme ℓ_1 minimale était une possibilité pour identifier des solutions parcimonieuses sans utiliser la norme ℓ_0. Il est ressorti de la discussion sur les indicateurs de parcimonie (p. 17) que la norme ℓ_p avec $0 < p < 1$ était aussi un bon indicateur. On s'intéresse donc à la possibilité de résoudre le problème (P_p) suivant :

$$(P_p) : \qquad \hat{x} = \arg\min_x \|x\|_p^p \text{ tel que } y = \Phi x \tag{2.5}$$

On remarque que la norme $\|x\|_p^p$ que l'on cherche à minimiser peut s'écrire $\sum_{i=1}^N |x_i|^{p-2} x_i^2$. On peut donc envisager le problème comme un problème de moindres carrés pondérés :

$$\min_x \|W^{-1}x\|_2^2 \qquad \text{tel que } \Phi x = y,$$

où W est une matrice. Si on place sur la diagonale de W^{-1} les éléments $|x_i|^{p/2-1}$, l'objectif devient bien la minimisation de la norme ℓ_p de la solution.

Lorsque W ne dépend pas de x, la solution s'écrit [GR97] $\hat{x} = W(W\Phi)^+ y$, où le $^+$ dénote la pseudo-inverse de Moore-Penrose. Mais ici, W dépend de x : il s'agit alors d'un problème non linéaire. On peut alors procéder de manière itérative, en fixant la

valeur de W à partir de la solution estimée $\hat{x}$ précédente, ce qui permet de calculer une nouvelle estimation de x et ainsi de suite. C'est ce que propose l'algorithme connu sous le nom de FOCUSS (*FOCal Underdetermined System Solver*) [GR97]. Cependant, lorsque x devient strictement parcimonieux, la matrice W devient singulière. Les auteurs de [GR97] proposent donc une régularisation de Tikhonov [Tik63] en résolvant le problème suivant :

$$\hat{x} = \arg\min_x \|y - \Phi x\|_2^2 + \lambda^2 \|Wx\|_2^2.$$

Le résultat devient alors

$$\hat{x} = (W^T \Phi^T \Phi W + \lambda I_N)^{-1} W^T \Phi^T y.$$

D'après [KDMR+03], cela peut se réécrire sous la forme

$$\hat{x} = W^T W \Phi^T (W^T \Phi \Phi^T W + \lambda I_N)^{-1} y = Q\Phi^T (\Phi Q \Phi^T + \lambda I_N)^{-1} y,$$

où $Q = W^T W = WW = W^T W^T = WW^T$.

Un autre algorithme appelé IRLS [CY08], pour *Iteratively Reweighted Least-Squares*, ressemble à FOCUSS. Cet algorithme cherche la solution de :

$$\min_x \sum_{i=1}^{N} w_i^{-2} x_i^2, \qquad \text{tel que } \Phi x = y, \tag{2.6}$$

où les poids sont tels que $w_i^{-1} = |x_i|^{p/2-1}$. On cherche là aussi à obtenir la solution de (P_p) par une méthode itérative : $w_{i|_n}^{-1} = |x_{i|_{n-1}}|^{p/2-1}$ où $x_{|_{n-1}}$ est l'estimée du signal à l'itération précédente. Lorsque l'algorithme converge, c'est-à-dire $x_{|_{n-1}} = x_{|_n}$, alors le critère à minimiser devient :

$$\sum_{i=1}^{N} w_{i|_n}^{-2} x_{|_n}^2 = \sum_{i=1}^{N} |x_{i|_{n-1}}|^{p-2} x_{i|_n}^2 = \sum_{i=1}^{N} |x_{i|_n}|^{p-2} |x_{i|_n}|^2 = \sum_{i=1}^{N} |x_{i|_n}|^p = \|x\|_p^p$$

Chaque itération est calculée à partir de la précédente, selon la formule suivante :

$$x \leftarrow Q\Phi^T (\Phi Q \Phi^T)^{-1} y, \tag{2.7}$$

où Q est la matrice diagonale ayant pour éléments les $1/w_i^2 = |x_i|^{2-p}$ lorsque $w_i \neq 0$. Comme on s'intéresse au cas où $0 \leq p \leq 1$, il faut prendre en considération le fait que les poids w_i ne sont plus définis si à un moment donné, $\exists x_i = 0$. La méthode de régularisation imaginée par [CY08] diffère légèrement de celle proposée pour FOCUSS (le but reste le même) : on ajoute un $\varepsilon > 0$ dans le poids :

$$w_i = \left(|x_{i|_{n-1}}|^2 + \varepsilon\right)^{p/2-1}. \tag{2.8}$$

En procédant ainsi, on peut garder les termes nuls de x dans la matrice Q (lorsque w_i est nul, il reste le terme en ε). Par conséquent, si une étape (notamment l'initialisation) fait apparaître un zéro là où il ne doit pas y en avoir, les résultats suivants ne sont pas limités au niveau du support de x. Ce terme de régularisation est variable : en effet, au début des itérations, on le choisit relativement élevé pour que son influence soit minime [CY08]. Ensuite, ce terme est diminué lorsque l'évolution entre deux itérations successives n'est plus suffisante, car on suppose que le signal devient parcimonieux et le recours à la régularisation devient nécessaire. On fixe un seuil minimal à ce paramètre afin de définir un critère d'arrêt pour l'algorithme. On remarque que ce seuil minimal conditionne en partie l'erreur de reconstruction. Selon les auteurs de [CY08], commencer avec un terme de régularisation important permettrait d'éviter de s'orienter vers un minimum local.

La figure 2.7 montre un exemple d'évolution de reconstruction d'un signal par l'algorithme. L'initialisation est choisie comme étant le résultat de la pseudo-inverse de Moore-Penrose : le signal n'a donc aucune raison d'être parcimonieux, et l'illustration montre qu'il ne l'est effectivement pas. Au fur et à mesure des itérations de l'algorithme, le signal estimé devient de plus en plus proche d'une solution parcimonieuse, jusqu'à converger vers le signal original. Comme il n'y a aucune assurance que l'algorithme converge vers le minimum global de l'équation (2.5), il peut être opportun de remplacer l'initialisation à l'aide de la pseudo-inverse de Moore-Penrose par quelques itérations de *Matching Pursuit* par exemple, afin d'augmenter la vitesse de l'algorithme et sa précision [BCI+07] en l'orientant vers une solution proche.

2.3 Conclusions

Pour clore ce chapitre, nous pouvons dire que le codage parcimonieux se résume à un ensemble codage/décodage et repose sur une hypothèse essentielle : le signal étudié est décomposable de manière parcimonieuse dans un espace de représentation (engendré par la matrice Ψ) connu ou qu'il est possible de déterminer (ce problème sera discuté au chapitre 6). L'aspect parcimonieux d'une représentation peut se quantifier de différentes manières. La plus naturelle est la norme ℓ_0, qui décompte le nombre d'éléments non nuls dans la représentation du signal. Cependant, avec des signaux réels, cette mesure risque souvent d'être inexploitable. Les normes ℓ_p constituent de bons indicateurs de parcimonie si $0 < p \leq 1$, et ne souffrent pas du même problème d'approximation du 0 que la mesure ℓ_0.

L'étape de codage est très simple : il s'agit d'une projection sur une matrice Φ, de dimension $m \times N$ avec $m < N$. Cette matrice ne doit pas être quelconque : il faut qu'elle respecte certains critères afin d'être efficace. Le critère principal est que la matrice Φ doit explorer toutes les colonnes du dictionnaire Ψ, c'est-à-dire qu'il ne doit pas être possible de décomposer parcimonieusement les lignes de Φ à l'aide des colonnes de Ψ, et récipro-

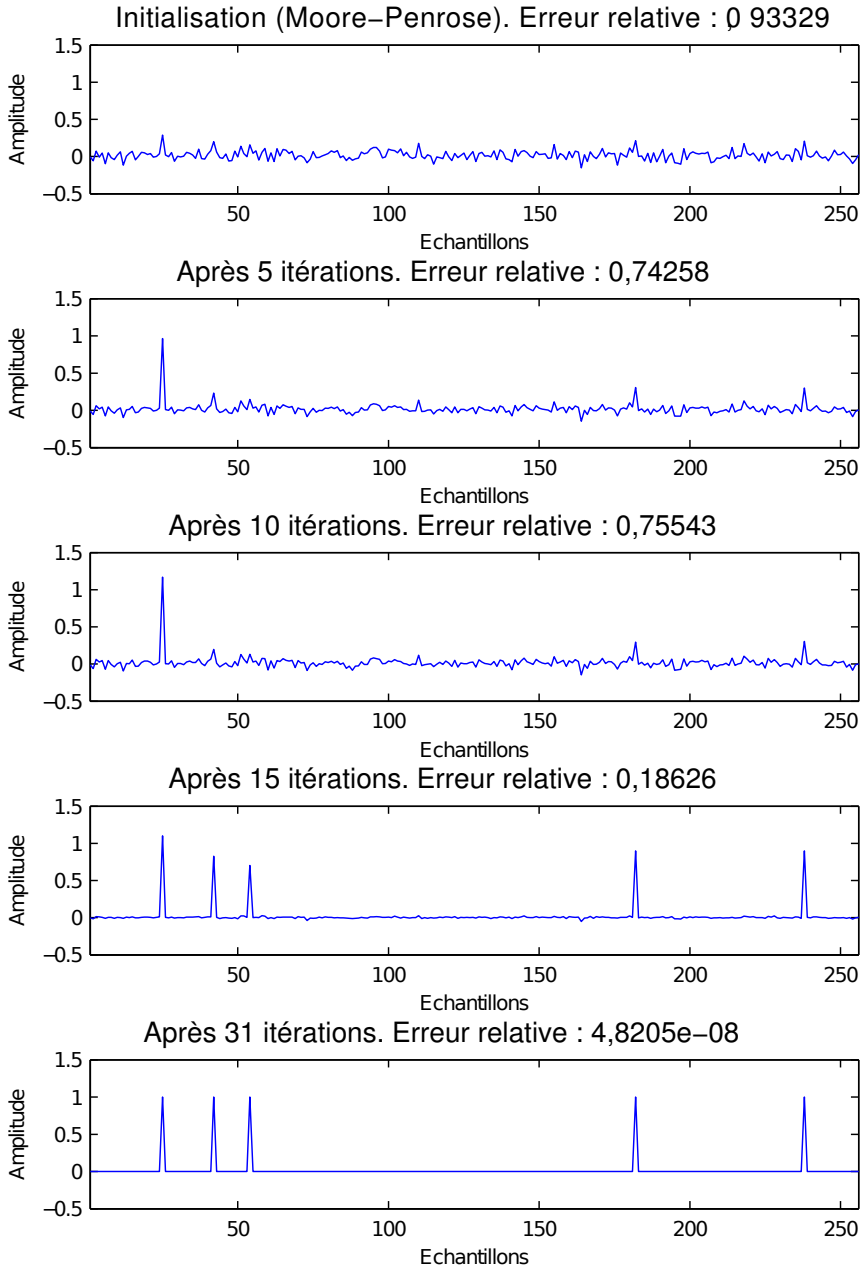

FIGURE 2.7 – Évolution de la reconstruction par l'algorithme IRLS d'un signal de longueur $N = 256$, nul partout sauf en 5 points, projeté sur une matrice aléatoire de taille $50 \times N$ (éléments issus d'un tirage aléatoire gaussien, i.i.d) dont les lignes ont été orthonormalisées.

quement. Cela se caractérise par un critère appelé cohérence, que l'on veut le plus bas possible. On retient essentiellement qu'une matrice issue d'un tirage aléatoire se montre généralement peu cohérente avec n'importe quel dictionnaire et est, de fait, parfaitement adaptée au codage parcimonieux.

C'est au niveau du décodage que l'hypothèse de parcimonie du signal entre en considération. Puisqu'on recherche une solution à un système pour lequel il en existe une infinité,

on choisit de chercher celle qui sera la plus parcimonieuse. Divers algorithmes existent pour cela, comme les algorithmes gloutons, qui construisent pas à pas une solution parcimonieuse, ou bien l'optimisation convexe qui donne la solution de norme ℓ_1 minimale, ou encore les moindres carrés pondérés visant la solution de norme ℓ_p minimale. Cette liste n'est pas exhaustive ([Nee09],[MBZJ09], ...) : ces algorithmes ont été considérés pour plusieurs raisons : historiquement, la minimisation ℓ_1 est aux origines du concept de codage parcimonieux, même si MP était utilisé avant pour faire de la décomposition parcimonieuse. Ensuite, parmi les nombreux autres algorithmes qui existent, un certain nombre sont des évolutions des algorithmes considérés ici. L'essentiel est que ces algorithmes permettent de reconstruire le signal d'origine, en connaissant la matrice de codage Φ et le dictionnaire Ψ de décomposition parcimonieuse du signal.

La question que l'on se pose maintenant est de savoir sur quels propriétés et résultats repose le concept de codage parcimonieux. Comment déterminer si la matrice permettra de retrouver une solution parcimonieuse ? À quelles conditions les divers algorithmes permettent-ils d'obtenir la bonne solution lorsque par exemple, on minimise la norme ℓ_1 plutôt que la norme ℓ_0 ?

Chapitre 3

Justifications théoriques du codage parcimonieux

Résumé : *Dans ce chapitre, nous abordons les résultats théoriques concernant les diverses approches de la reconstruction dans le codage parcimonieux. Notamment, les conditions d'exactitude de la reconstruction, de la relaxation en norme ℓ_1 du problème (P_0), ainsi que des bornes lorsque le signal n'est pas strictement parcimonieux.*

Dans le chapitre précédent, on a décrit le principe du codage parcimonieux en détail, en évoquant quelques conditions à suivre pour que cela fonctionne. On retient notamment l'hypothèse fondamentale qui est la possibilité de décomposer le signal de manière parcimonieuse. Il est aussi question de la matrice de codage, souvent choisie aléatoirement parce que "ça marche bien". C'est le cas parce que ces matrices vérifient certaines propriétés avec de très fortes probabilités. Enfin, on a discuté quelques algorithmes de reconstruction qui répondent à des formulations diverses du problème. Dans ce chapitre, on récapitule les résultats présents dans la littérature, qui permettent d'établir les conditions requises sur le signal et la matrice de codage pour s'assurer d'une reconstruction exacte du signal parcimonieux, ou pour quantifier l'erreur de reconstruction si celle-ci n'est pas exacte.

3.1 Problème de reconstruction

Rappelons le problème : l'observation y est une projection du signal $x \in \mathbb{R}^N$ sur la matrice $\Phi \in \mathbb{R}^{m \times N}$ avec $m < N$. On suppose dans ce chapitre que x est parcimonieux dans la base canonique de $\mathbb{R}^N$ ($\Psi = Id_N$), c'est-à-dire $\|x\|_0 \leq k << N$. Les résultats présentés s'appliquent à la matrice d'observation Φ. Dans le cas où le signal se décompose

de manière parcimonieuse dans un dictionnaire autre que la base canonique, l'ensemble des résultats doit être considéré en remplaçant Φ par le produit $\Phi\Psi$.

Pour reconstruire le signal, on considère dans un premier temps le problème (P_0) suivant :

$$(P_0): \qquad \hat{x} = \arg\min_x \|x\|_0 \text{ tel que } y = \Phi x \tag{3.1}$$

Afin de donner des garanties sur la solution de (P_0), la notion de *spark* est introduite par [DE03].

Définition 9 (Spark d'une matrice) *Soit une matrice Φ. On définit $\sigma = \text{spark}(\Phi)$ comme le plus petit nombre tel qu'il existe un sous-ensemble de σ colonnes de Φ linéairement dépendantes.* [1]

Cette valeur est au moins égale à 2 si la matrice n'a pas de colonne nulle. De manière assez évidente, le rang de la matrice permet de déterminer une valeur maximale du spark : $\text{spark}(\Phi) \leq \text{rang}(\Phi) + 1$. Il est facile de construire une matrice de rang m dont le spark vaut 2 en partant d'une matrice de rang m et en y adjoignant une colonne identique à l'une de celles déjà présentes.

Un premier résultat concernant l'exactitude de la solution de (P_0) est le suivant :

Théorème 1 *[DE03, BZJ10] Si le problème (P_0) accepte une solution $\hat{x}$ telle que $\|\hat{x}\|_0 < \frac{1}{2}\text{spark}(\Phi)$, alors cette solution est unique.*

En effet, si on considère deux solutions distinctes u et v alors $d = u - v$ appartient au noyau de Φ : $\Phi d = 0$. Cela veut dire que les $\|d\|_0$ colonnes de Φ sélectionnées par d sont linéairement dépendantes et donc que $\|d\|_0$ est au moins égal au à $\text{spark}(\Phi)$ par définition, donc $\|u - v\|_0 \geq \sigma$. Or, $\|u - v\|_0 \leq \|u\|_0 + \|v\|_0$ et donc $\|u\|_0 + \|v\|_0 \geq \text{spark}(\Phi)$. Il ne peut donc y avoir qu'une seule solution $\hat{x}$ telle que $\|\hat{x}\|_0 < \frac{1}{2}\text{spark}(\Phi)$.

Par hypothèse, l'observation y est construite à partir du signal x et de sa projection sur Φ. Si x vérifie $\|x\|_0 < \frac{1}{2}\text{spark}(\Phi)$, alors le théorème 1 assure que si la résolution de (P_0) conduit à une solution $\hat{x}$ telle que $\|\hat{x}\|_0 < \frac{1}{2}\text{spark}(\Phi)$, alors celle-ci est exacte : $x = \hat{x}$.

Dans l'optique de formuler d'autres garanties de reconstruction de x par (P_0), la notion de *propriété d'isométrie restreinte* (RIP, en anglais *restricted isometry property*) est introduite par [Can08, CT05]. Cette propriété caractérise la tendance d'une matrice à se comporter comme une isométrie pour des vecteurs parcimonieux (au sens de la norme ℓ_0), au travers de la *constante d'isométrie restreinte* δ_k associée :

1. Une autre manière de définir le spark est la suivante :

$$\text{spark}(\Phi) = \min_{x \in \text{Ker}\,\Phi,\, x \neq 0} \|x\|_0$$

Définition 10 (Constante d'isométrie restreinte) *Soit une matrice $\Phi \in \mathbb{R}^{m\times N}$. Pour tout $k \leq N$, on définit la constante d'isométrie restreinte à k de la matrice Φ comme la plus petite valeur $\delta_k > 0$ telle que, pour tout $x \in \mathbb{R}^N$ tel que $\|x\|_0 \leq k$,*

$$(1-\delta_k)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta_k)\|x\|_2^2. \tag{3.2}$$

k est un majorant de $\|x\|_0$ et n'est pas nécessairement entier ni égal au nombre de composantes non nulles du signal.

Si la constante δ_k est nulle, la matrice se comporte comme une isométrie pour les vecteurs comportant au plus k éléments non nuls, tandis que si la constante est inférieure à 1, cela signifie qu'il n'y a pas de vecteur à au plus k composantes non nulles dans le noyau de Φ : la projection Φx n'est jamais nulle pour ces vecteurs. Si $\delta_{2k} < 1$, on a alors le résultat suivant :

Théorème 2 *[Can08] Si $\delta_{2k} < 1$, alors le problème (P_0) a une unique solution $\hat{x}$ telle que $\|\hat{x}\|_0 = k$.*

En effet, s'il existe deux solutions distinctes k-parcimonieuses u et v, alors leur différence $d = u - v$ appartient au noyau de Φ : $\|\Phi d\|_2 = \|\Phi(u-v)\|_2 = \|\Phi u - \Phi v\|_2 = \|y - y\|_2 = 0$. Comme il y a deux solutions distinctes à k éléments non nuls, leur différence d a au plus $2k$ éléments non nuls, et donc d'après l'hypothèse $\delta_{2k} < 1$, on a (éq. 3.2) : $\|\Phi d\|_2^2 \geq (1-\delta_{2k})\|d\|_2^2 > 0$, ce qui est en contradiction avec $\|\Phi d\|_2 = 0$.

On remarque que la condition du théorème 2 ci-dessus sur la constante d'isométrie restreinte δ_{2k} d'une matrice est liée au *spark* de celle-ci. Nous proposons le résultat suivant :

Proposition 1 *Soit une matrice $\Phi \in \mathbb{R}^{m\times N}$, si Φ vérifie (3.2) avec $\delta_{2k} < 1$ alors $k < \frac{\text{spark}(\Phi)}{2}$*

En effet, si $\delta_{2k} < 1$, pour tout x tel que $0 < \|x\|_0 \leq 2k$, $\|\Phi x\|_2^2 \geq (1-\delta_{2k})\|x\|_2^2 > 0$, et par conséquent, il n'existe pas d'ensemble de $2k$ colonnes (ou moins) liées de Φ, c'est-à-dire, $\text{spark}\,\Phi > 2k$. La réciproque n'est pas vraie : la matrice $2 * Id_N$ (où Id_N est la matrice identité de taille N) est un contre-exemple trivial : prenons $N \geq 2$ et le vecteur $x = [1\,1\,0\,0\,...\,0] \in \mathbb{R}^N$: $\|x\|_2 = \sqrt{2}$ et $\|x\|_0 = 2 < \text{spark}\{2 * Id_N\} = N+1$, $\|2 * Id_N x\|_2 = 2\sqrt{2}$, donc si $\delta_{2k} < 1$, par l'équation (3.2), on a : $\|2 * Id_N x\|_2 = 2\sqrt{2} \leq (1+\delta_{2k})\|x\|_2 < 2\|x\|_2 = 2\sqrt{2}$: il y a contradiction.

On a donc un moyen de s'assurer que la reconstruction du signal est exacte si la matrice d'observation suit les propriétés énoncées ci-dessus, dès lors que le signal estimé est suffisamment parcimonieux.

3.1.1 Relaxation convexe

Ces premiers résultats (th. 1 et 2) sont intéressants, mais il y a un obstacle conséquent : résoudre le problème (P_0) n'est pas réalisable en temps polynomial car c'est un problème NP-difficile [Nat95]. Il faudrait énumérer toutes les solutions possibles de parcimonie k : il y en a $\binom{N}{k}$. Il a donc été proposé [CDS01, CRT06a] de remplacer (P_0) (éq. 3.1) par *Basis Pursuit*, comme il a été mentionné au chapitre précédent :

$$(P_1): \qquad \hat{x} = \arg\min_x \|x\|_1 \text{ tel que } y = \Phi x, \tag{3.3}$$

Cette relaxation conduit à un problème convexe, que l'on sait résoudre. Les résultats expérimentaux [CDS01] illustrent la possibilité de reconstruire un signal parcimonieux par la résolution de *Basis Pursuit* ((P_1)). Le cas de la relaxation ℓ_1, appliquée à l'exemple de la figure 2.3 où l'on projette un signal constitué de quelques pics temporels sur m lignes de la matrice de Fourier discrète choisies au hasard, est justifié par le théorème suivant :

Théorème 3 *[CRT06a] Si on connait m coefficients de Fourier d'un signal $x \in \mathbb{C}^N$ composé de k pics temporels (i.e. on connait $y = F_m x$ où F_m est une matrice $m \times N$ issue de la matrice de Fourier $N \times N$,) et que*

$$k \leq C_m \frac{m}{\log N},$$

alors avec une probabilité $1 - O\left(N^{-m}\right)$, la solution de (P_1) est exactement x.

Les résultats théoriques de [DH02, Don06] montrent que ceci est vrai aussi pour d'autres types de matrices et que la minimisation ℓ_1 retrouve les solutions parcimonieuses.

3.1.1.1 Constante d'isométrie restreinte et relaxation convexe

Dans la section précédente, nous avons introduit la constante d'isométrie restreinte (éq. 3.2). Celle-ci est reprise dans les théorèmes justifiant l'exactitude de la relaxation convexe. Un premier théorème démontre que la relaxation convexe peut être exacte :

Théorème 4 *[CRT06b] Si $\|x\|_0 \leq k$ et si la matrice Φ est telle que*

$$\delta_{3k} + 3\delta_{4k} < 2,$$

alors x est la solution unique de (P_1).

Un second théorème ultérieur affine ce résultat :

Théorème 5 *[Can08] Si $\delta_{2k} < \sqrt{2} - 1$, alors la solution $\hat{x}$ au problème (P_1) vérifie les résultats suivants :*

$$\|\hat{x} - x\|_1 \leq C_0 \|x - x_{\mathbf{k}}\|_1$$

et

$$\|\hat{x} - x\|_2 \leq C_0 \frac{1}{\sqrt{k}} \|x - x_{\mathbf{k}}\|_1,$$

où

$$C_0 = 2\frac{1 - (1 - \sqrt{2})\delta_{2k}}{1 - (1 + \sqrt{2})\delta_{2k}},$$

et $x_{\mathbf{k}}$ est la restriction de x aux k composantes les plus importantes.

La conséquence intéressante de ce théorème est que si le vecteur est k-parcimonieux, au sens où $\|x - x_{\mathbf{k}}\|_2 = 0$, alors la reconstruction est exacte ; mais s'il ne l'est pas, l'erreur est bornée.

Le problème du bruit n'est pas évoqué dans ces résultats, mais une variante de *Basis Pursuit* appelée *Basis Pursuit DeNoising (BPDN)* traite le problème de la mesure bruitée, c'est à dire

$$y = \Phi x + w,$$

où $w \in \mathbb{R}^m$ est un terme de bruit inconnu tel que $\|w\|_2 \leq \varepsilon$. Le problème devient alors

$$(BPDN) : \qquad \hat{x} = \arg\min_x \|x\|_1 \text{ tel que } \|y - \Phi x\|_2 \leq \varepsilon. \tag{3.4}$$

On a alors le résultat suivant :

Théorème 6 *[Can08] Si $\delta_{2k} < \sqrt{2} - 1$ et $\|w\|_2 \leq \varepsilon$, alors la solution $\hat{x}$ de (3.4) vérifie :*

$$\|\hat{x} - x\|_2 \leq C_0 \frac{1}{\sqrt{k}} \|x - x_{\mathbf{k}}\|_1 + C_1 \varepsilon,$$

où $x_{\mathbf{k}}$ est la restriction de x aux k composantes les plus importantes, C_0 définie comme au théorème 5 et

$$C_1 = 2\frac{4\sqrt{1 + \delta_{2k}}}{1 - (1 + \sqrt{2})\delta_{2k}}.$$

On retrouve donc un résultat similaire à celui du théorème 5, tout en bornant l'influence du bruit sur le résultat.

Cependant, il faut remarquer que les constantes C_0 et C_1 divergent :

$$\lim_{\substack{\delta_{2k} \to \sqrt{2}-1 \\ \delta_{2k} < \sqrt{2}-1}} C_0 = +\infty ;$$

$$\lim_{\substack{\delta_{2k} \to \sqrt{2}-1 \\ \delta_{2k} < \sqrt{2}-1}} C_1 = +\infty.$$

Le critère $\delta_{2k} < \sqrt{2} - 1$ permet donc d'obtenir des bornes sur l'erreur, mais dans la pratique, ces bornes sont très élevées. La figure 3.1 illustre les valeurs prises par les deux constantes, et on remarque que pour $\delta_{2k} > 0.3$, les valeurs de ces constantes deviennent

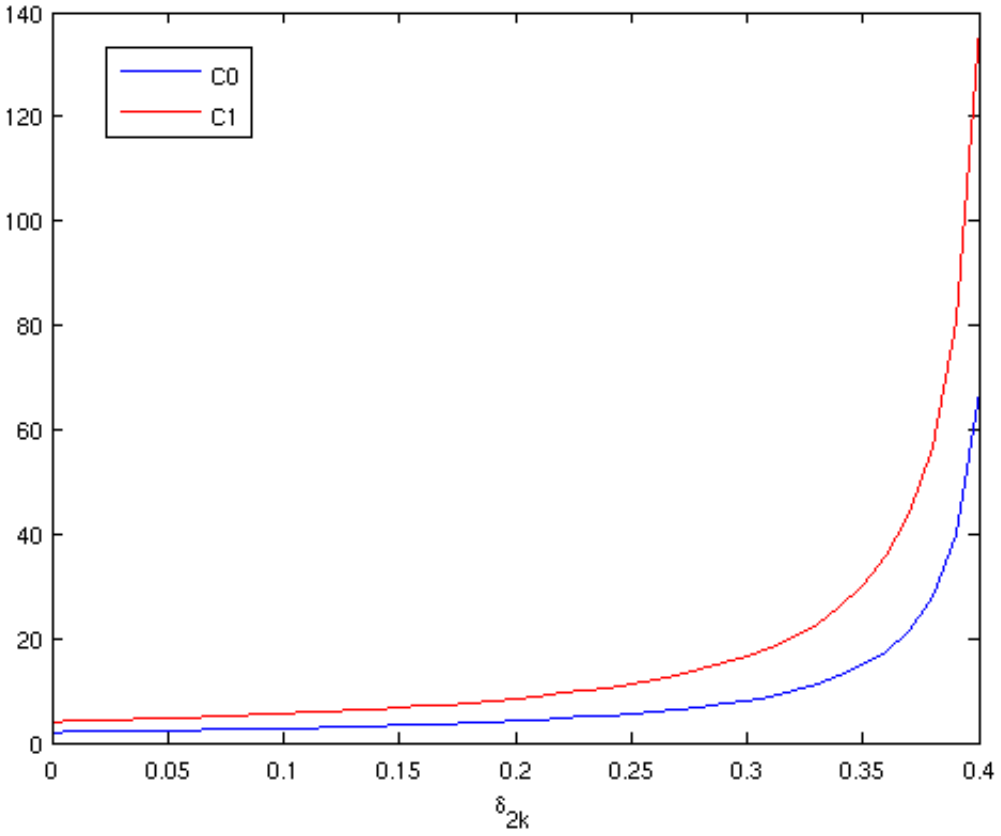

FIGURE 3.1 – Évolution des constantes C_0 et C_1 des théorèmes 5 et 6 en fonction de la valeur de δ_{2k}.

très importantes. Dans la pratique où les signaux ne sont généralement pas strictement parcimonieux au sens de la norme ℓ_0 (ne serait-ce qu'à cause du bruit), il ne faut donc utiliser que des matrices dont la constante d'isométrie est faible si l'on veut être assuré d'une bonne reconstruction. Si on est trop proche de la limite $\sqrt{2}-1$, l'erreur possible, conditionnée par les constantes C_0 et C_1, devient trop importante.

3.1.1.2 Cohérence de la matrice d'observation

Un autre critère souvent présenté dans la littérature pour qualifier (au sens du codage parcimonieux) une bonne matrice d'observation Φ est sa cohérence, définie p. 39, qui correspond au maximum de corrélation entre ses colonnes. En effet, si les colonnes de la matrice sont faiblement corrélées, il est intuitivement plus facile de déterminer la combinaison linéaire de colonnes composant l'observation, et donc de retrouver le signal original. Cette intuition vient facilement si l'on pense aux algorithmes gloutons, qui recherchent les colonnes de la matrice les plus corrélées avec le signal : si deux colonnes sont trop proches (donc fortement corrélées) il est difficile de déterminer exactement laquelle contribue au

signal compressé.

Définition 11 (Cohérence) *Soit une matrice $\Phi \in \mathbb{R}^{m \times N}$, la cohérence M de cette matrice est définie comme le maximum de corrélation entre ses colonnes.*

$$M = \max_{\substack{1 \leq i,j \leq N \\ i \neq j}} \frac{< |\Phi_i, \Phi_j| >}{\|\Phi_i\|_2 \|\Phi_j\|_2}$$

Cette mesure M permet de définir une propriété d'équivalence entre les solutions des problèmes (P_1) (éq. 2.3) et (P_0) (éq. 2.3) :

Théorème 7 *[EB02] Si Φ est la concaténation de deux base orthonormées, et si $\|\hat{x}\|_0 \leq 1/M$ alors c'est l'unique solution de (P_0). Si en plus, $\|\hat{x}\|_0 \leq \frac{1+M^{-1}}{2}$ alors c'est aussi l'unique solution de (P_1).*

[EB02] ont ensuite amélioré ce résultat en ramenant la valeur limite de $\|\hat{x}\|_0$ à $\frac{\sqrt{2}-0,5}{M}$ pour l'équivalence entre (P_1) et (P_0).

Cependant, ces propriétés manquent de généralité, puisque l'on traite uniquement des dictionnaires constitués d'une paire de base orthonormées.

3.1.2 Conclusions

La relaxation en norme ℓ_1 permet de donner un sens au codage parcimonieux. Des garanties sur l'exactitude de cette relaxation sont données en s'appuyant sur les caractéristiques de la matrice. La plus célèbre est la propriété d'isométrie restreinte. Cette dernière indique qu'une matrice a tendance à préserver les vecteurs parcimonieux. Pour que la matrice soit efficace, il ne faut pas que de petits ensembles de colonnes soient liés (c'est l'idée du spark), la constante d'isométrie restreinte ajoute une finesse supplémentaire. Cependant, le calcul de ces constantes (pour différentes valeurs de k) est fastidieux (il faut considérer les $\binom{N}{k}$ sous-ensembles possibles d'un nombre donné k de colonnes). Heureusement, les matrices aléatoires vérifient cette propriété avec une très forte probabilité [BDDW08]. Donc les matrices aléatoires, en plus d'être un codeur "universel" (cf. section 2.2.1), vérifient les propriétés justifiant la relaxation en norme ℓ_1.

Toutefois, on a vu que la résolution de (P_1) n'était pas la seule méthode de reconstruction possible. Nous revenons maintenant sur les autres méthodes mentionnées au chapitre précédent.

3.1.3 Algorithme de poursuite

Tropp et Gilbert [TG07] ont démontré que la reconstruction avec Orthogonal Matching Pursuit d'un signal $x \in \mathbb{R}^N$ tel que $\|x\|_0 = k$ à partir de $m \propto k \ln N$ mesures par projections linéaires, était exacte, et ce avec une probabilité élevée :

Théorème 8 *[TG07] Soit $\beta \in [0, 0.36]$. Si la matrice $\Phi \in \mathbb{R}^{m \times N}$ est issue d'un tirage aléatoire gaussien, si $\|x\|_0 = k$, et si $m \geq Ck \ln \frac{N}{\beta}$, alors l'algorithme OMP reconstruit x à partir de $y = \Phi x$ avec une probabilité supérieure à $1 - 2\beta$.*

La constante vérifie $C \leq 20$ et peut être réduite pour les grandes valeurs de k [TG07].

La garantie n'est ici pas aussi forte que celle du théorème 6 : pour une matrice aléatoire donnée, la reconstruction est seulement assurée avec une forte probabilité, alors que pour le théorème 6, la reconstruction est assurée dès que la matrice vérifie la propriété d'isométrie restreinte. Cependant, on voit dans la suite de ce chapitre (p.41) qu'une matrice aléatoire vérifie la propriété d'isométrie restreinte dans des conditions, et avec une probabilité, similaires à celles du théorème 8 ci-dessus.

3.1.4 Relaxation non-convexe

Dans les différents résultats précédents, on constate que les garanties sur la relaxation convexe du problème (P_0) sont plus strictes que celles pour le problème lui-même. Notamment, un exemple donné par [BZJ10] montre la différence entre les conditions régissant (P_0) et celles régissant l'équivalence avec (P_1). Si on construit un dictionnaire en concaténant une base de Dirac et une base de Fourier $\Phi = [Id_m, F_m]$, alors on a $N = 2m$. Dans ce cas là, la cohérence M prend la valeur $M = \frac{1}{\sqrt{m}}$ [GN03]. Il est évident que $\text{spark}(\Phi) = m + 1$. Si on choisit $N = 1000$, alors le théorème 1 indique qu'un vecteur x tel que $\|x\|_0 \leq 250$ est nécessairement l'unique solution du problème. Le théorème 7, quant à lui, garantit que si la solution de (P_1) est telle que $\|\hat{x}\|_0 < \sqrt{m}(\sqrt{2} - 0.5) = 20,44$, alors elle est exacte. Donc, avec une telle matrice Φ, pour un vecteur k parcimonieux, tel que $21 \leq k \leq 250$, on ne peut garantir, en s'appuyant sur la cohérence, que la relaxation convexe soit exacte. Au niveau des constantes d'isométrie restreinte, la seule condition relative à la solution de (P_0) est que $\delta_{2k} < 1$. Pour que la relaxation ℓ_1 soit exacte, il faut $\delta_{2k} < \sqrt{2} - 1$, ce qui est plus contraignant que $\delta_{2k} < 1$, qui impose uniquement que tout sous-ensemble de $2k$ colonnes forme une famille libre.

Les normes ℓ_0 et ℓ_1 sont deux mesures de parcimonie, mais ne sont pas les seules. Les normes ℓ_p, $0 < p < 1$ ont aussi été évoquées précédemment. Nous nous intéressons alors au problème suivant :

$$(P_p) : \qquad \hat{x} = \arg\min_x \|x\|_p \text{ tel que } y = \Phi x \tag{3.5}$$

Contrairement à (P_1), cette relaxation n'est pas convexe : nous avons vu (section 2.2.2) que sa résolution n'est pas aussi immédiate et assurée que dans le cas convexe.

Une généralisation du théorème 4 (p.36) est proposée par [Cha07b] :

Théorème 9 *Soient $0 < p \leq 1$, $b > 1$ et $a = b^{\frac{p}{2-p}}$. Si $\|x\|_0 \leq k$ et que la matrice Φ est telle qu'elle vérifie la propriété d'isométrie restreinte (éq. 3.2) avec les constantes δ telles que*

$$\delta_{ak} + b\delta_{(a+1)k} < b - 1,$$

alors x est la solution unique de (P_p).

Pour $b = 3$ et $p = 1$, on retrouve le résultat du théorème 4. Le fait que a ne soit pas nécessairement entier n'est pas incompatible avec la définition 10 de la constante d'isométrie restreinte, puisque ak, $a(k+1)$ sont des majorants de la norme ℓ_0.

Un résultat intéressant est un corollaire de ce dernier théorème :

Théorème 10 *[Cha07a] Si $\delta_{2k+1} < 1$ et $p > 0$, alors x est l'unique solution du problème (P_p).*

La restriction est alors moins contraignante que pour la relaxation convexe, la condition étant très proche de celle du théorème d'unicité de la solution de (P_0).

3.2 Discussion

Un certain nombre des résultats évoqués dans ce chapitre dépendent de la matrice d'observation et de sa constante d'isométrie restreinte δ_k (éq. 3.2). La valeur de cette constante permet de donner des garanties d'unicité de la solution et de borner l'erreur de reconstruction due au bruit ou à une approximation parcimonieuse, mais il faut relativiser son utilisation : calculer la valeur de la constante pour une matrice donnée est fastidieux, car il faut considérer pour calculer δ_k tous les sous-ensembles de k colonnes [2] et en calculer les valeurs singulières. Ceci constitue un nouvel avantage à l'utilisation de matrices aléatoires : le théorème 11 ci-dessous permet d'affirmer qu'avec une forte probabilité, une matrice dont les éléments sont issus d'un tirage aléatoire gaussien possède une constante d'isométrie restreinte faible, permettant d'appliquer les théorèmes énoncés en début de chapitre. Cela est un avantage de plus pour les matrices aléatoires, qui sont statistiquement incohérentes avec toute matrice déterministe et dont la cohérence entre les colonnes est statistiquement faible. Quel que soit le critère, les matrices aléatoires le remplissent avec une forte probabilité.

Théorème 11 *[BDDW08] Soit une matrice $\Phi \in \mathbb{R}^{m \times N}$ dont les éléments sont issus d'un processus aléatoire gaussien centré de variance $\frac{1}{m}$. Soit $0 < \delta < 1$, donné, alors il existe deux constantes strictement positives c_1, c_2 telles que Φ vérifie la RIP (def. 10) :*

$$(1 - \delta)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta)\|x\|_2^2,$$

2. Il y en a C_N^k.

avec la constante δ *et tout* $k \leq c_1 \frac{m}{\log \frac{N}{k}}$ *avec une probabilité supérieure à* $1 - 2e^{-c_2 m}$.

Rappelons aussi que si Φ vérifie la RIP pour un couple (k, δ_k) donné, et si Ψ est unitaire, alors $\Phi\Psi$ vérifie aussi la propriété d'isométrie restreinte avec la même constante δ_k, mais si Ψ n'est pas unitaire, il n'est pas assuré que le produit vérifie la propriété.

Cependant, d'après la formule proposée par [ME09a] pour connaitre m lorsque l'on fixe δ_k, k, N et la probabilité minimale voulue $p_{\min}$ avec laquelle une matrice de type vérifiera la RIP avec cette constante δ_k, pour de petites valeurs de N, le résultat n'est pas exploitable. Par exemple, si $N = 256$, pour avoir $\delta_2 < \sqrt{2} - 1$ avec $p_{\min} = 0.95$, il faut $m = 258$, pour une matrice aléatoire à deux valeurs symétriques équiprobables (± 1). Dans le chapitre suivant, on présente des simulations qui montrent que résoudre (P_1) fonctionne pour des valeurs de m plus faibles et k plus élevées. La valeur de la constante permet d'établir une condition suffisante pour assurer une reconstruction exacte, mais pas nécessaire.

Des méthodes comme *StRIP*, pour statistical RIP [CHJ10], et *ExRIP*, pour Expected RIP [ME09a], proposent une estimation de la probabilité qu'une matrice respecte la propriété d'isométrie restreinte pour une constante δ_k et un k donnés. Ceci permet de déterminer avec quelle probabilité le type de matrice vérifiera la propriété d'isométrie restreinte, et donc d'assurer une reconstruction exacte en résolvant *Basis Pursuit* (éq. 3.3).

Puisque les conditions de reconstruction apparaissent très strictes et difficiles à remplir, ou données parfois avec des constantes non définies, des simulations devraient permettre d'avoir une meilleure appréciation des situations où le codage parcimonieux est applicable.

Chapitre 4

Résultats expérimentaux

Résumé : *Ce chapitre présente des résultats expérimentaux sur des exemples simples de signaux strictement parcimonieux. On y compare les algorithmes présentés au chapitre 2 : algorithmes gloutons (MP, OMP), minimisation ℓ_1 et moindres carrés pondérés. On s'intéresse également au choix de la matrice aléatoire de codage, en observant les résultats pour des matrices à valeur réelles ou binaires. Enfin, on observe une application à des signaux enregistrés au cours d'une campagne expérimentale.*

Dans les chapitres précédents, on a décrit le codage parcimonieux de manière théorique, notamment sur les conditions permettant d'assurer l'exactitude de la reconstruction. On se propose maintenant de chiffrer ces conditions à partir d'exemple simples, en simulation. Ces simulations permettent de comparer les différentes méthodes de reconstruction présentées en section 2.2.1 du chapitre 2, ainsi que différentes matrices de codage. Cela permet aussi de considérer deux autres problèmes non traités jusqu'ici : le choix de la matrice de codage, et l'algorithme de reconstruction. Enfin, un exemple sur des signaux enregistrés, issus d'expériences visant à caractériser la réponse d'une dalle en béton à divers évènements, permet de juger des capacités du codage parcimonieux dans une situation plus réelle.

4.1 Simulation sur des signaux synthétiques

4.1.1 Méthodes

Pour ces simulations, nous avons considéré les algorithmes suivants :

- Matching Pursuit (MP) [DDWB06] ;
- Orthogonal Matching Pursuit (OMP) [TG07] ;
- Iteratively Reweighted Least Squares (IRLS), avec $p = 0$ (cf. paragraphe 4.1.2.1) [CY08] ;
- l1eq_pd, de la bibliothèque ℓ_1-MAGIC, qui résout le problème (P_1) (éq. 2.3) [CR05].

Leur implémentation est détaillée en annexe B.

Afin de comparer empiriquement les performances de ces algorithmes, des séries de tests ont été effectuées sur des signaux synthétiques, parcimonieux dans la base canonique ($\Psi = I_d$). La figure 4.1 illustre les trois cas envisagés. Dans chaque cas, la position des k éléments non nuls est choisie aléatoirement, mais les valeurs prises par ces éléments diffèrent :

- (a) : l'amplitude prend aléatoirement, de manière équiprobable, les valeurs $+1$ ou -1 ;
- (b) : l'amplitude est la réalisation d'un tirage aléatoire gaussien centré de variance 1 ;
- (c) : l'amplitude vaut $+1$.

Le critère retenu pour comparer la qualité de reconstruction des algorithmes, c'est-à-dire la capacité à reconstruire un signal, est l'erreur relative en norme ℓ_2, $\|\hat{x} - x\|_2 / \|x\|_2$. Celle-ci doit être inférieure à 10^{-3}, ce seuil ayant été déterminé empiriquement à partir de l'observation de la distribution de l'erreur. En pratique, pour les signaux bien reconstruits, l'erreur est bien inférieure à ce seuil (cf. annexe B).

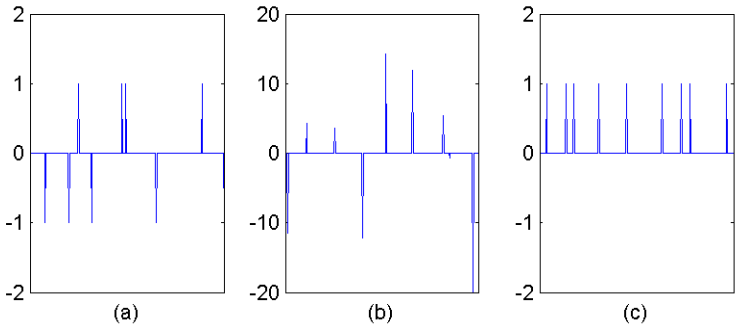

FIGURE 4.1 – Exemples des trois types de signaux parcimonieux synthétiques utilisés dans les simulations ($k = 9$, $N = 256$).

4.1.2 Comparaison des algorithmes

La figure 4.2 montre une comparaison des performances des quatre algorithmes dans le cas des signaux synthétiques de type (a) décrits dans la section précédente. Cette comparaison repose sur le pourcentage de signaux reconstruits en fonction de la parcimonie k du signal et du nombre d'observations m, en utilisant comme matrices de projection des matrices aléatoires gaussiennes. On constate que l'algorithme MP est bien moins performant que les autres dans le sens où, à parcimonie k fixée, il faut plus d'observations (m plus grand) pour reconstruire correctement le signal. Les algorithmes IRLS et l1eq_pd de ℓ_1-MAGIC ont des performances similaires, alors que celles d'OMP sont légèrement en deçà : à partir de $k = 8$, $m = 64$ observations ne sont plus suffisantes pour reconstruire le signal.

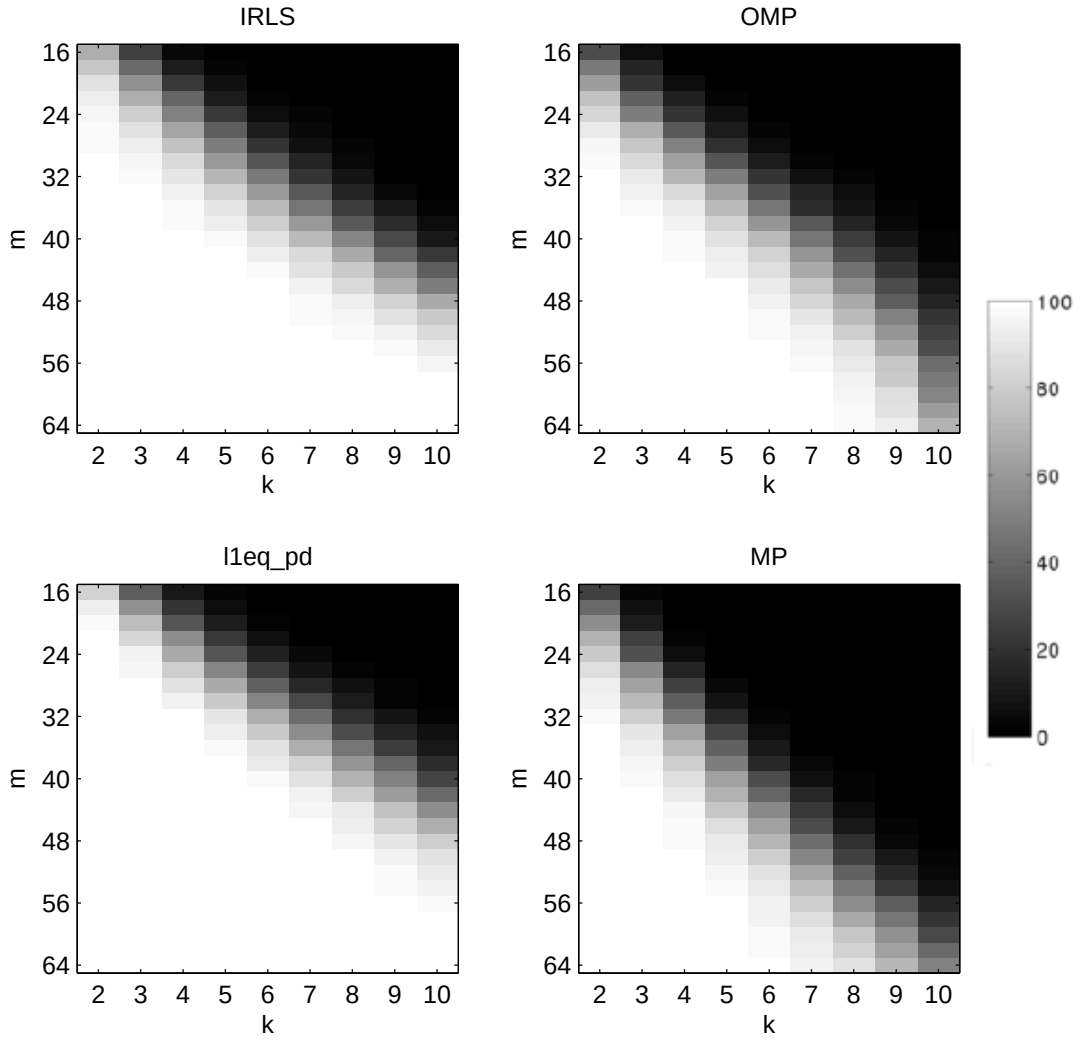

FIGURE 4.2 – Comparaison des performances de reconstruction en fonction de k et m pour les algorithmes IRLS (a), OMP (b), l1eq_pd (c) et MP (d). Les signaux sont du type (a) sur la figure 4.1, $N = 256$. En blanc, 100% des signaux sont reconstruits avec une erreur relative inférieure à 10^{-3}, en noir, aucun n'est reconstruit.

La figure 4.3 montre les performances des 4 algorithmes selon le type de signal observé, comme décrit en figure 4.1. On remarque que la performance de la minimisation de norme ℓ_1 n'est pas sensible au type de signal observé, alors que pour les trois autres algorithmes, un signal où les amplitudes sont aléatoires est mieux reconstruit. Dans ce cas là, IRLS est le plus performant. On remarque à nouveau l'écart de performance qui sépare l'algorithme Matching Pursuit des autres.

La figure 4.4 montre les performances de reconstruction des quatre algorithmes pour des signaux de parcimonie $k = 6$, pour les trois types de signaux présentés en figure 4.1. On constate que dans chacun des trois cas, le nombre d'observations nécessaires pour avoir 100 % de reconstructions exactes (pour notre critère) avec IRLS est inférieur à celui requis

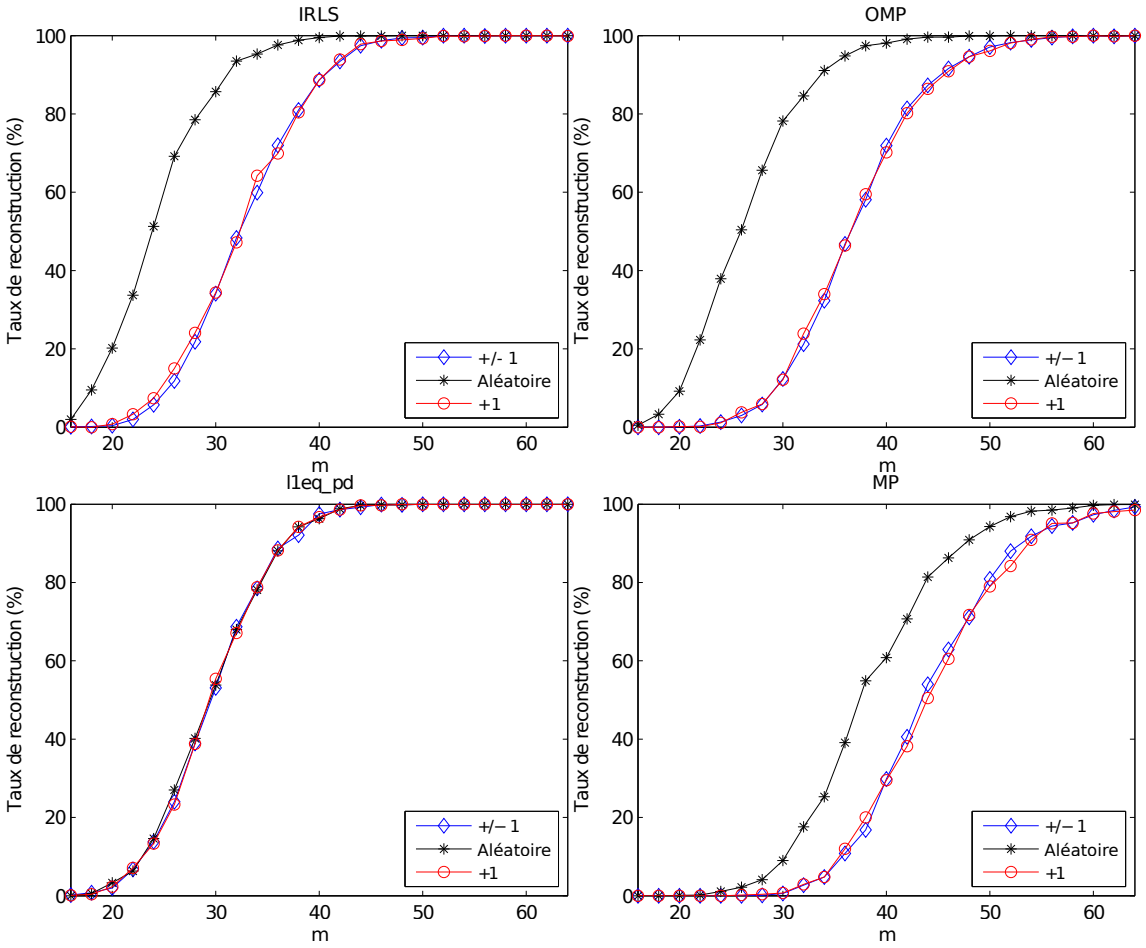

FIGURE 4.3 – Comparaisons pour chaque algorithme, des performances de reconstruction pour des signaux de taille $N = 256$ et de norme ℓ_0 $k = 6$, pour les trois types de signaux synthétiques (cf. fig. 4.1.) La matrice de projection est aléatoire gaussienne à lignes orthonormées.

pour OMP, mais qu'avec ce même nombre d'observation, OMP reconstruit quand même plus de 95% des signaux. De manière similaire, l'algorithme l1eq_pd est plus performant qu'IRLS pour les signaux où les grandeurs sont ± 1, avec là aussi, au moins 95% des signaux reconstruits par IRLS quand 100% le sont avec l1eq_pd. Il ne faut cependant que quelques observations de plus pour atteindre le taux de 100% avec IRLS. En revanche, pour des signaux parcimonieux où les amplitudes sont aléatoires (type (b)), les performances avec l1eq_pd sont moins bonnes qu'avec OMP, mais la différence n'est là aussi que de moins de 5 observations. En revanche, obtenir 100% de signaux reconstruits avec Matching Pursuit nécessite systématiquement, sur ces simulations, au moins 10 observations de plus qu'avec les autres algorithmes.

Par la suite, on privilégiera donc les algorithmes IRLS ou l1eq_pd. Cependant, choisir l'algorithme OMP peut être intéressant dans les cas où l'on veut facilement préciser le nombre maximal de composantes à trouver, même si l'approximation est erronée. Ceci permet aussi de réduire le coût calculatoire.

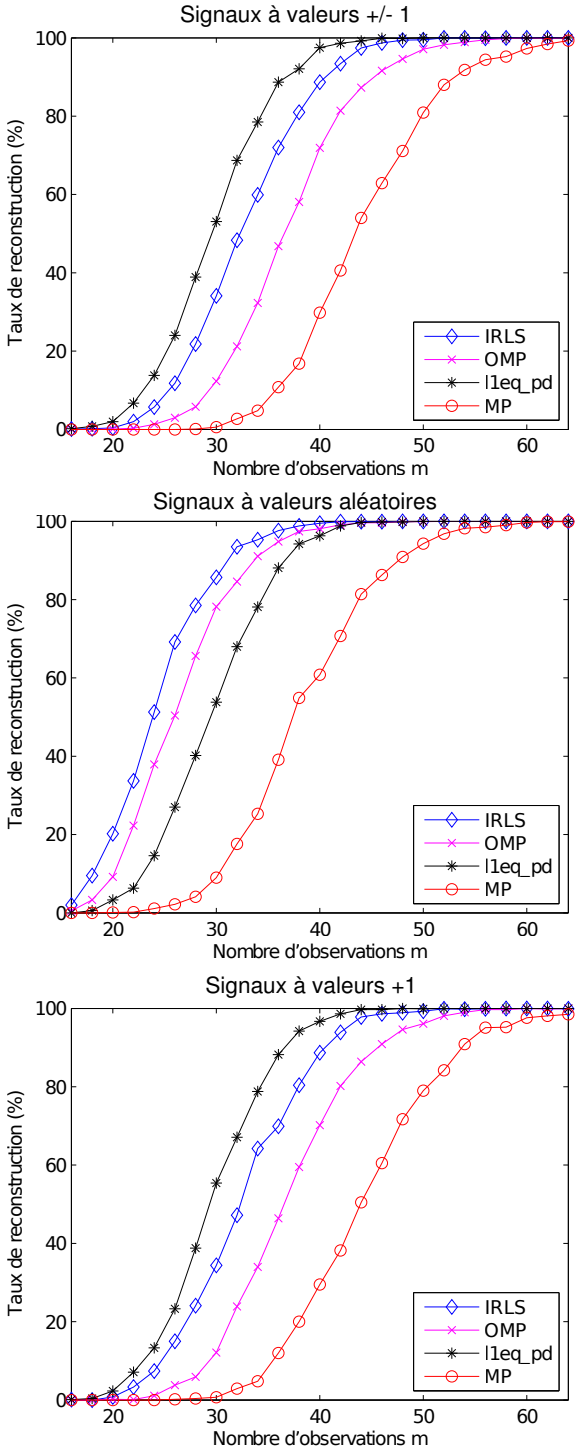

FIGURE 4.4 – Comparaison des 4 algorithmes pour $N = 256$, $k = 6$ et m variant de 15 à 65 pour des signaux parcimonieux des trois types présentés par la figure 4.1, dans le même ordre (± 1, amplitude aléatoire, et $+1$.

4.1.2.1 Paramétrisation de p dans l'algorithme IRLS

L'algorithme IRLS permet de choisir la norme ℓ_p que l'on cherche à minimiser. Expérimentalement [Cha07a, Cha07b] il vaut mieux chercher à minimiser une norme ℓ_p avec p proche de zéro pour obtenir une bonne solution, car les conditions d'exactitude de la relaxation de (P_0) (éq. (2.2)) en (P_p) (éq. (2.5)), et donc d'unicité de la solution, sont moins difficiles à remplir lorsque p est petit [CS08] (cf. p.40.) Pour illustrer ceci, des simulations sur des signaux de type (c) (fig. 4.1) avec des matrices aléatoires gaussiennes dont les lignes ont été orthonormalisées, ont été réalisées : la figure 4.5 illustre le fait que réduire p pour approcher 0 ne permet pas nécessairement une meilleure reconstruction. En effet, alors qu'on constate une nette amélioration des performances lorsque p varie entre 1 et 0,7, celles-ci stagnent une fois $p < 0,5$. Cependant, la figure 4.6 montre que le nombre d'itérations nécessaires pour converger vers le résultat est plus petit lorsque p est plus proche de 0. On retient donc qu'il est avantageux d'utiliser une valeur de $p < 0,5$ et qu'utiliser $p = 0$ n'est pas handicapant, du moins sur les signaux strictement parcimonieux de ces simulations.

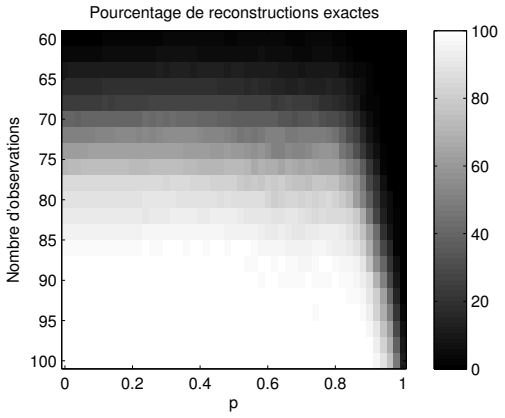

FIGURE 4.5 – Pourcentage de signaux dont l'erreur relative en norme ℓ_2 de reconstruction est inférieure à 10^{-3}, pour l'algorithme IRLS, en fonction de p variant de 0 à 1 et du nombre d'observations m variant de 60 à 100. Les signaux sont de taille $N = 256$ et leur norme ℓ_0 vaut 20.

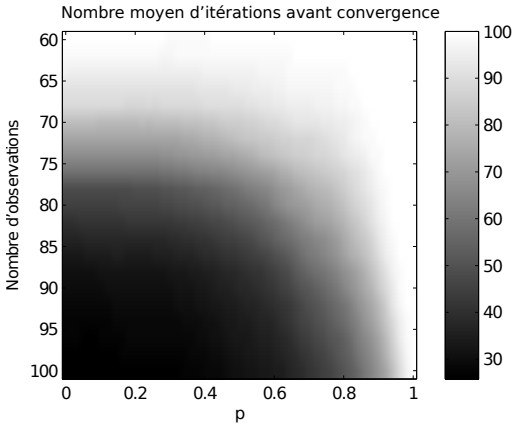

FIGURE 4.6 – Nombre moyen d'itérations pour atteindre une erreur relative en norme ℓ_2 de reconstruction inférieure à 10^{-3}, pour l'algorithme IRLS, en fonction de p variant de 0 à 1 et du nombre d'observations m variant de 60 à 100. Les signaux sont de taille $N = 256$ et leur norme ℓ_0 vaut 20. Le nombre d'itérations est limité à 100 (on considère que ce seuil est suffisamment large pour permettre la convergence lorsque c'est possible).

4.1.3 Codage : les diverses matrices d'observation

Il ressort des discussions précédentes que les matrices aléatoires gaussiennes sont adaptées à une utilisation comme matrice d'observation "universelle" dans un système de codage

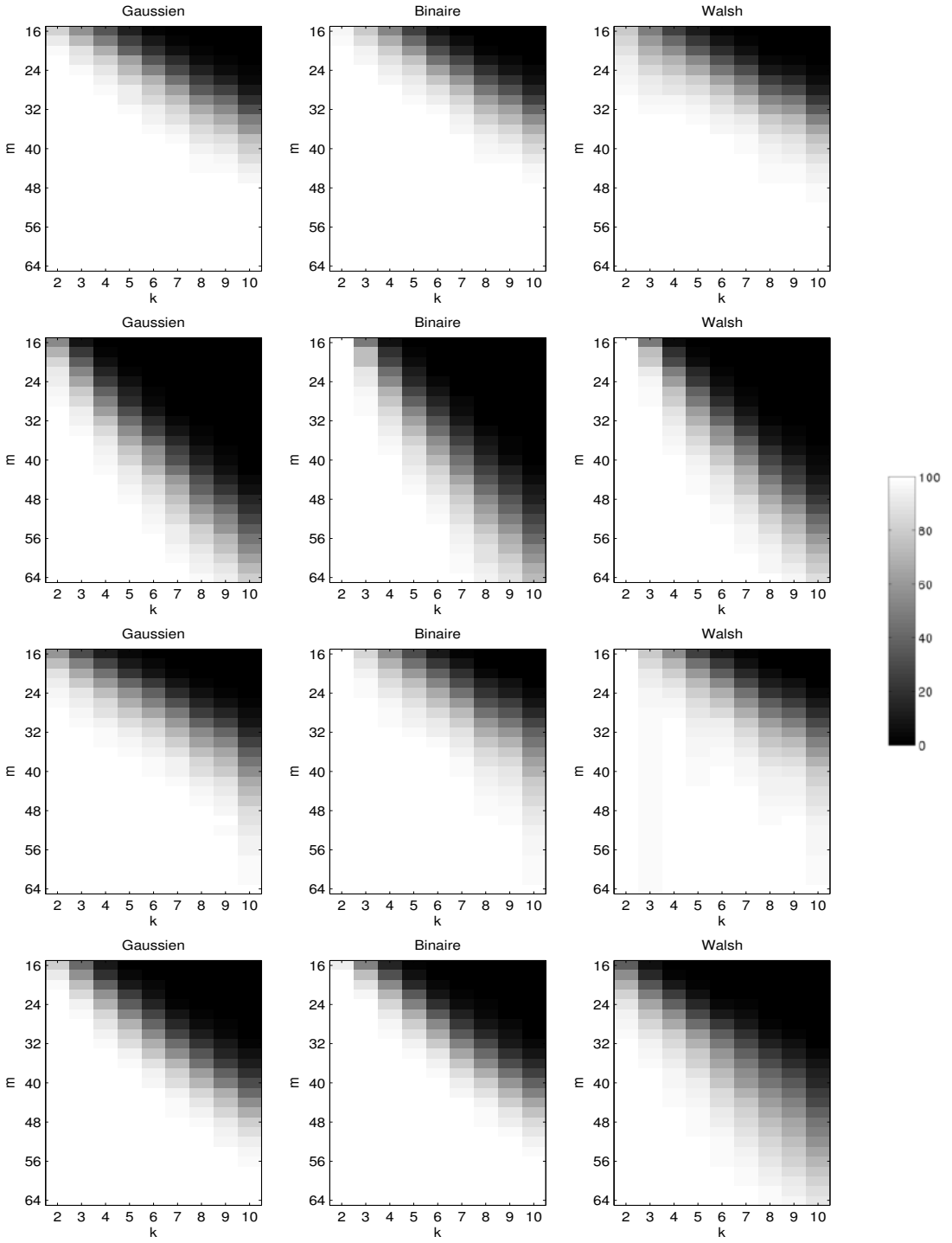

FIGURE 4.7 – Comparaison des performances de reconstruction selon k et m pour des matrices aléatoires gaussiennes (gauche), des matrices aléatoires binaires (centre) et les matrices binaires de type Walsh (droite), avec, de haut en bas, les algorithmes IRLS, MP, OMP, l1eq_pd. Les signaux utilisés sont du type (b) (fig. 4.1, $N = 256$). En blanc, 100% des signaux sont reconstruits avec une erreur relative inférieure à 10^{-3}, en noir, aucun n'est reconstruit.

parcimonieux. Les matrices aléatoires binaires, à valeurs ± 1 équiprobables, ont aussi été envisagées puisque ces dernières permettent de simplifier l'implémentation du codage par-

cimonieux (cf. chapitre 5). Statistiquement, ces séquences sont orthogonales entre elles : ceci nous assure qu'elles explorent au maximum l'espace de départ, puisqu'elles forment une famille libre. Si on cherche à avoir des séquences strictement orthogonales, on peut s'intéresser aux séquences utilisées en communication par les techniques d'étalement de spectre (*spread spectrum*) : ces séquences sont construites de manières à ce qu'elles soient toutes orthogonales entre elles. Les matrices de Walsh-Hadamard en sont un exemple, parmi d'autres (séquences de Gold [Gol68], Kasami [Kas66], etc). Une alternative possible à la matrice aléatoire binaire est donc de choisir m lignes [1] dans une matrice de Walsh-Hadamard de taille N. Ce choix peut être fait de manière aléatoire. Cela a l'avantage de fournir une matrice de rang m , alors que cela n'est vrai que d'un point de vue statistique pour une matrice aléatoire. Cependant, la méthode de construction des matrices de Walsh-Hadamard impose que N soit une puissance de 2, ce qui peut éventuellement être une contrainte.

Dans cette section, on évalue donc l'influence d'un tel choix en comparaison avec celui d'une matrice aléatoire gaussienne sur les performances du codage parcimonieux, sur des exemples de signaux synthétiques idéaux (fig 4.1).

Les performances, illustrées par la figure 4.7 sont surprenantes dans le sens où, les matrices binaires étant moins générales, il aurait semblé logique que les performances soient moins bonnes. Pour les signaux fortement parcimonieux ($k \leq 2$,) il faut moins d'observations pour reconstruire les signaux qu'avec une matrice aléatoire gaussienne. La différence est cependant peu sensible, l'écart diminue lorsque k augmente et l'avantage revient aux matrices gaussiennes lorsque $k \geq 8$. Les différences de performance entre les matrices binaires aléatoires et les matrices de Walsh-Hadamard sont faibles, excepté pour le cas de MP : les performances sont bien meilleures qu'avec les matrices aléatoires, notamment lorsque $k > 5$.

On retient surtout qu'utiliser une matrice binaire n'empêche pas le fonctionnement du codage parcimonieux, du moins sur ces exemples, et les performances sont comparables à celles produites lors d'un codage par une matrice aléatoire gaussienne. On s'attend cependant à une incompatibilité avec des signaux en forme de créneaux (décomposition parcimonieuse en ondelettes de Haar.)

4.2 Signaux expérimentaux réels

Jusqu'à présent, nous avons simulé des signaux synthétiques idéaux puisque strictement parcimonieux dans la base canonique de $\mathbb{R}^N$ ($\Psi = Id_N$), et nous avons validé le principe du codage parcimonieux. Cependant, il serait intéressant de savoir comment cela s'applique à une situation réelle, dans laquelle les signaux ne sont plus parcimonieux dans le temps mais dont on suppose qu'il existe une manière de les représenter parcimonieusement par

1. On évite la première ligne, qui n'est pas de moyenne nulle.

rapport à une base ou un dictionnaire Ψ que l'on connait. Pour cela, nous nous sommes intéressés à des signaux de vibration, dont la nature impulsionnelle et transitoire rend le choix du dictionnaire bien moins évident.

4.2.1 Description des signaux

Les signaux utilisés sont issus d'une campagne de mesures [2] et sont obtenus à l'aide d'accéléromètres placés au sol enregistrant divers événements comme la chute d'une bille (premier rebond uniquement), un coup de marteau ou un pas de marche. La figure 4.8 montre un exemple de la réponse enregistrée à la chute d'une bille. Pour ces signaux, il faut déterminer une base de représentation permettant une décomposition parcimonieuse. Le caractère impulsionnel du signal écarte la transformée de Fourier discrète de ce rôle. La figure 4.9 montre, si besoin en était, que la TFD de ce signal ne peut pas être qualifiée de parcimonieuse. En substitut, on se propose d'utiliser des bases d'ondelettes discrètes. Un choix empirique se porte sur des ondelettes de type Symmlet à 10 moments, dont la forme s'apparente à ce qui est observable sur les signaux. De plus, on constate expérimentalement que ce type là d'ondelette permet d'obtenir les décompositions les plus parcimonieuses (au sens des normes ℓ_p, $p < 1$) par rapport à une sélection d'ondelettes choisies pour leur ressemblance physique avec le signal (Daubechies (8, 10, 12, 14 moments), Symmlet (10, 14, 18 moments).

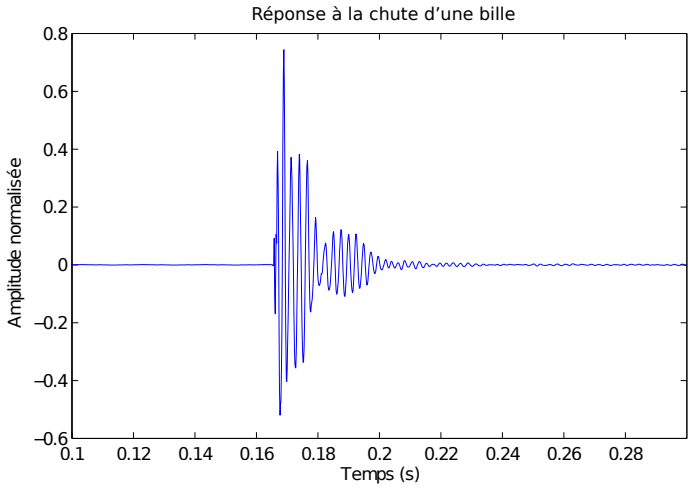

FIGURE 4.8 – Réponse à la chute d'une bille sur une dalle de béton, échantillonnée à $F = 6400$ Hz.

4.2.2 Performance

Dans cette expérience, nous mettons en évidence la capacité à coder le signal et à le reconstruire avec des erreurs limitées. La figure 4.11 (gauche) montre les résultats obtenus

2. Remerciements à M. Carmona, R. Bahroun, O. Michel pour ces signaux.

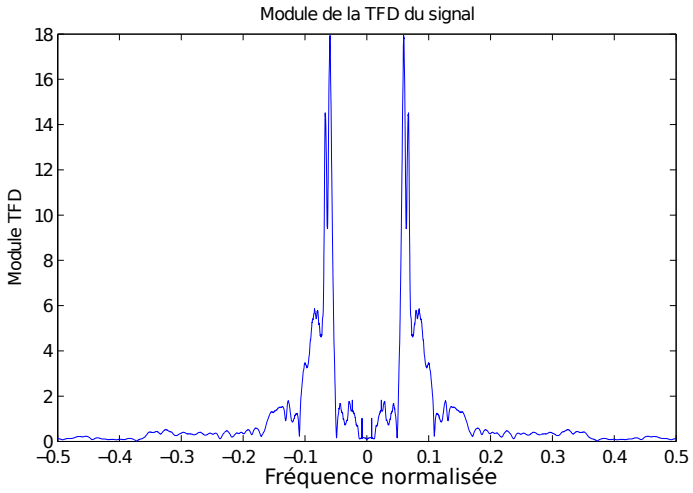

FIGURE 4.9 – Module de la transformée de Fourier discrète du signal observé sur la figure 4.8.

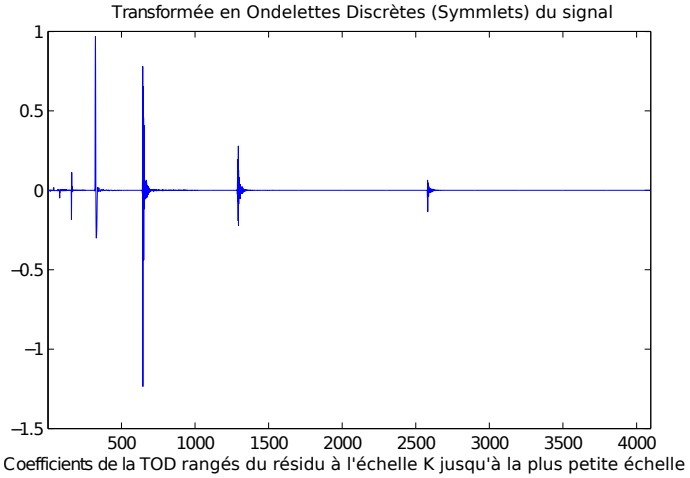

FIGURE 4.10 – Transformée en ondelettes discrète, avec ondelette de type Symmlet, du signal observé sur la figure 4.8. Les coefficients sont ordonnés de la résolution la plus grossière à la plus fine.

pour un exemple d'une chute de bille à proximité du capteur et pour laquelle le rapport signal sur bruit est élevé tandis que la figure de droite montre une moyenne effectuée sur divers signaux obtenus pour plusieurs événements enregistrés par des capteurs plus ou moins proches du point d'impact (c'est-à-dire avec un RSB plus ou moins bon). On constate qu'en réduisant d'un facteur 10 la taille des données, l'erreur relative est contenue (inférieure à 10%,) et inférieure pour les matrices de type aléatoire par rapport aux matrices de type Walsh. Toutefois, atteindre une telle erreur (moins de 10%) ne requiert que quelques coefficients, si on effectue une transformée en ondelettes discrète sur ces signaux (ce qui est plus lourd en calcul qu'une simple projection sur une matrice binaire), tandis que pour un taux de compression équivalent (facteur 10), l'erreur de reconstruction est nulle ou négligeable. Cependant, il semble qu'il ne soit pas possible d'obtenir une reconstruction

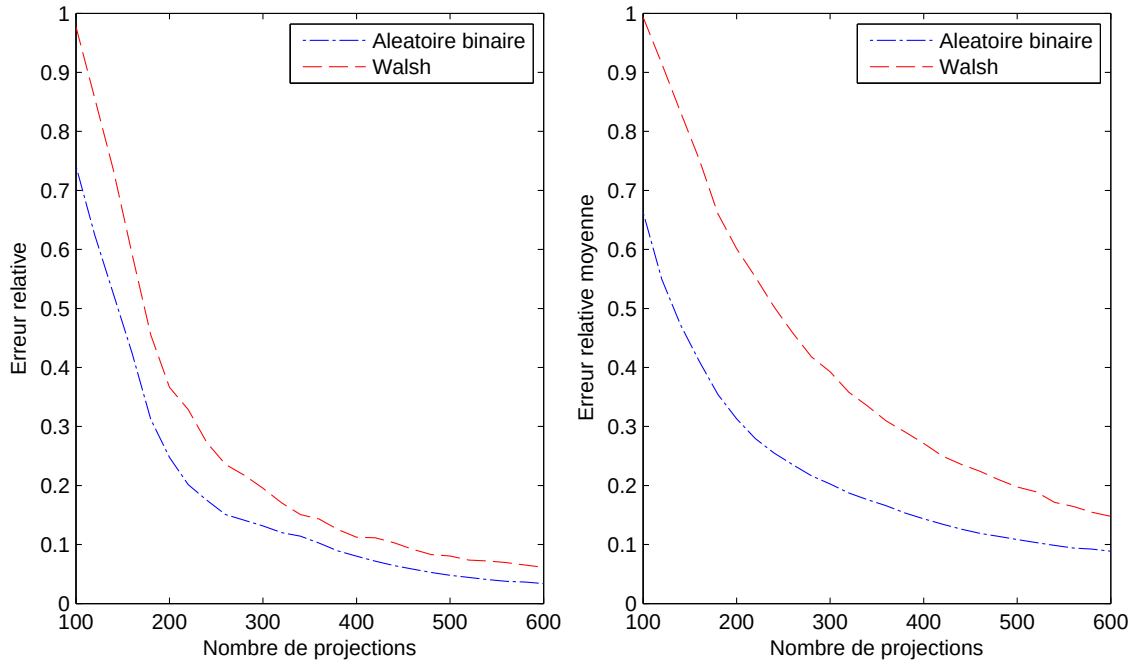

FIGURE 4.11 – Erreur relative de reconstruction en norme ℓ_2 moyenne en fonction de m. La matrice d'observation est soit aléatoire binaire, soit issue d'une matrice de Walsh-Hadamard, de taille $m \times N$ avec $N = 4096$. À gauche, le signal observé est la chute d'une bille, illustrée par la figure 4.8 ; à droite, c'est une moyenne obtenue avec divers signaux de type chute de bille, coup de marteau, pas, à des distances variables du capteur (le RSB décroit lorsque la distance augmente).

exacte. Ceci est certainement dû au choix de la matrice de représentation parcimonieuse qui n'est pas parfaitement adaptée aux signaux. On constate visuellement sur la figure 4.12, représentant le détail d'un signal reconstruit après un codage parcimonieux, qu'une partie de l'erreur provient d'artefacts situés à l'extérieur de la zone significative et sur la figure 4.13, qui est un grossissement de la précédente, que la reconstruction apparaît fidèle lorsqu'on considère la zone significative.

4.3 Conclusion

Les simulations effectuées dans ce chapitre permettent d'avoir une idée des performances qui peuvent être attendues du codage parcimonieux. On retient que le choix de matrice aléatoire semble convaincant, et surtout, le fait de choisir des matrices à état binaire apparaît tout aussi efficace, ce qui représente un gros avantage au moment de réfléchir à une implantation : les opérations, déjà peu coûteuses, sont encore réduites puisqu'on peut travailler sur des valeurs entières. Les essais sur les algorithmes mettent en avant les lacunes de l'algorithme Matching Pursuit, dont l'unique avantage reste celui du coût de calcul (qui se ferait mieux ressentir sur des exemples de plus grandes dimensions). Les performances d'OMP sont nettement meilleures que celles de MP et se rapprochent for-

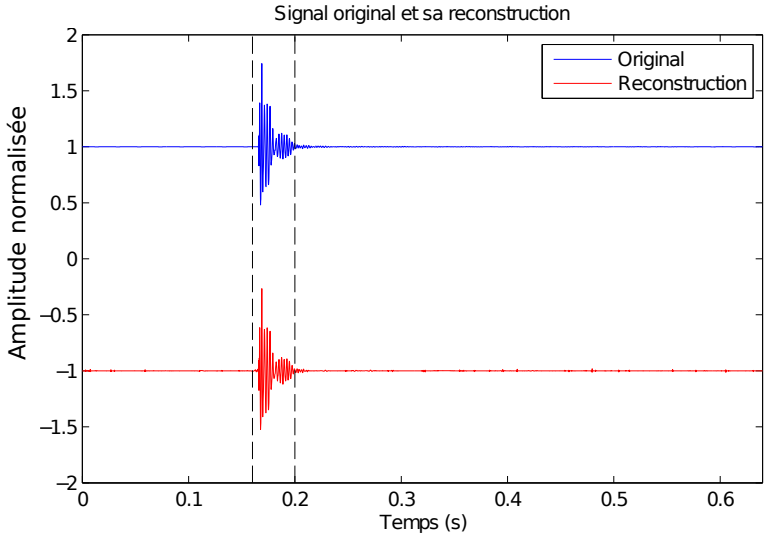

FIGURE 4.12 – Vue d'un signal original et du signal reconstruit, pour $m = 400$, $N = 4096$ et une matrice aléatoire binaire. L'erreur relative de reconstruction en norme ℓ_2 vaut 0.0976

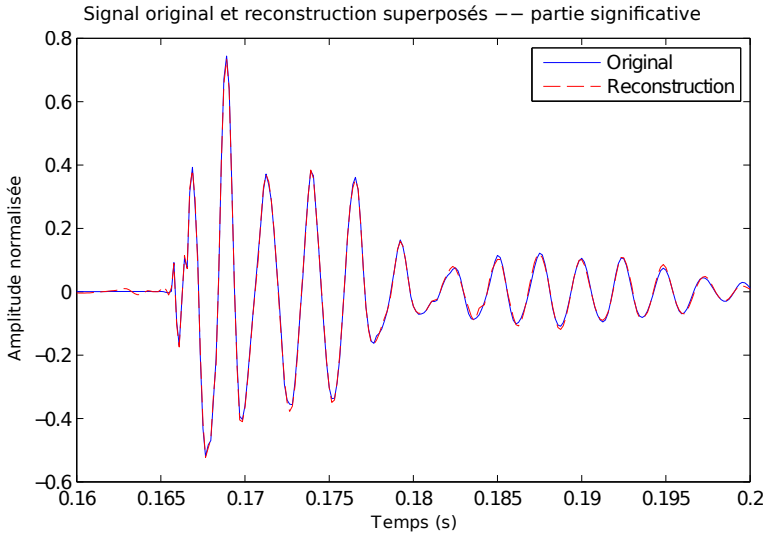

FIGURE 4.13 – Grossissement sur la partie significative (délimitée par les pointillés verticaux) du signal de la figure 4.12.

tement de celles des algorithmes d'optimisation convexe et IRLS. OMP a aussi l'avantage de permettre le choix a priori du nombre de composantes à reconstruire si l'on ne veut qu'une approximation et de ne pas faire plus de calculs que nécessaires. Tels qu'ils sont conçus et implémentés, les algorithmes IRLS et d'optimisation convexe imposent de calculer l'ensemble des composantes du signal.

Les expérimentations sur les signaux réels illustrent la difficulté de choisir un dictionnaire de représentation qui permet la décomposition parcimonieuse du signal. Les méthodes traditionnelles (Fourier, ondelettes) ne sont pas toujours adaptées. Les résultats restent mo-

destes, tant du point de vue du taux de compression atteint, que de l'erreur affichée. Cela met en évidence le besoin de déterminer des méthodes de représentation adaptées, car plus on a des décompositions parcimonieuses, plus le codage est efficace.

Chapitre 5

Application du codage parcimonieux à un démonstrateur

Résumé : *Dans ce chapitre, nous abordons les aspects de la réalisation d'un démonstrateur utilisant les techniques de codage parcimonieux afin de réduire le débit de données d'un capteur. L'objectif est d'augmenter l'autonomie de capteurs sans-fil en diminuant les transmissions.*

Nous nous intéressons dans ce chapitre à la mise en place d'un système permettant le codage du signal par projection sur une matrice. La partie décodage a été traitée précédemment et on suppose donc qu'elle ne pose plus de problèmes : on fait l'hypothèse que l'on sait recevoir les données et les traiter de manière logicielle pour reconstruire le signal.

L'objectif visé ici est la conception d'un système simple permettant de faire l'acquisition et la compression de signaux micro-sismiques, comme l'exemple de la figure 4.8 (p.51.) Ces signaux peuvent être échantillonnés facilement de manière traditionnelle car les fréquences observées sont inférieures à 5 kHz. Cependant, dans le contexte d'un réseau de capteurs de grande taille, les débits de données deviennent difficiles à gérer du point de vue de la transmission (notamment, la consommation et le spectre disponible pour les transmissions sans fil) et du stockage. Le but est donc de réduire le débit de données résultant de l'acquisition de ces signaux. Les simulations effectuées en 4.2 ont montré qu'il était raisonnable d'envisager une réduction par un facteur 10 du débit de données. Le second objectif, connexe au premier, est la simplicité du système, pour un coût et une consommation réduits (notamment grâce à la réduction du débit de données) pour s'adapter à de larges réseaux de capteurs autonomes, pour lesquels les ressources en énergie sont faibles.

Deux approches sont possibles ici :

– le convertisseur analogique-information (CAI, ou *A2I* pour *Analog to Information*) permettant de passer directement du signal analogique à l'observation numérique

dans le domaine compressé, voir figure 5.1. L'échantillonnage est alors effectué après la projection sur la matrice de codage Φ, et donc à fréquence réduite.

– Une méthode de compression par le codage parcimonieux, appliqué à un échantillonnage classique de Shannon, qu'on appellera *codeur numérique parcimonieux (CNP)*, voir figure 5.2.

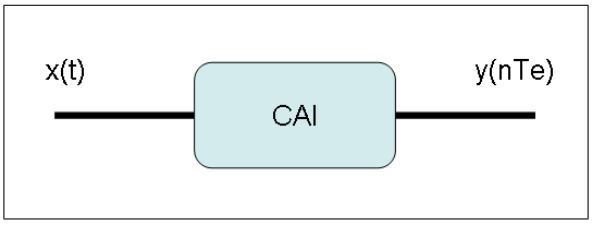

FIGURE 5.1 – Convertisseur Analogique-Information.

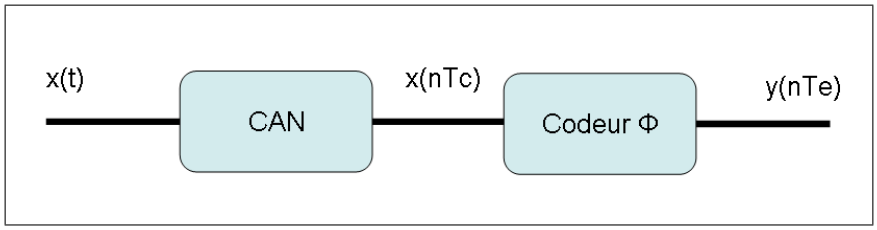

FIGURE 5.2 – Compression parcimonieuse après échantillonnage de Shannon.

La première approche est présente dans de nombreux exemples de la littérature [LKD+07, KLW+07, MEDS09, TLD+09, RKL+08, ME10] qui traitent principalement de l'échantillonnage de signaux ultra-large bande [1] (ULB, en anglais *UWB*,) pour lesquels l'échantillonnage numérique classique est difficile, voire impossible, à réaliser. Le CAI permet de contourner cette difficulté en appliquant le principe du codage parcimonieux pour "*battre Shannon*" en réduisant la fréquence de fonctionnement des échantillonneurs (d'un facteur $\frac{N}{m}$ à N selon si l'implémentation utilise 1 ou m échantillonneur(s)). Cependant, cela nécessite de générer des séquences pseudo-aléatoires correspondant aux lignes de la matrice de codage, cadencées à la fréquence d'échantillonnage nécessaire pour couvrir le spectre du signal et de produire le mixage du signal avec ces séquences. Ceci entraine des contraintes de réalisation, notamment du point de vue de la consommation, qui ne sont pas justifiées pour les problèmes envisagés dans ce document puisque sont traités des signaux dont l'échantillonnage ne pose pas de problème de réalisation. Une étude des possibilités de convertisseur analogique-information a toutefois été réalisée et est présentée en annexe C de ce document.

Dans la seconde approche, le signal est échantillonné normalement à la fréquence nécessaire après un filtrage passe-bas, puis projeté numériquement sur la matrice Φ de taille $m \times N$. On retrouve là le mode de fonctionnement envisagé dès le début de ce document, au travers des divers exemples du chapitre 2. Le facteur de réduction du débit de données attendu est donc $\frac{N}{m}$. On choisit d'utiliser comme matrice d'observation une matrice binaire,

1. Signaux RF : UHF Ultra Haute Fréquence 300MHz à 3GHz, SHF Super Haute Fréquence 3GHz à 30GHz

prenant les valeurs ± 1. Ceci s'explique par la volonté de simplicité de la solution : il suffit de changer le bit de signe de l'échantillon. [2]

5.1 Codeur numérique parcimonieux

Pour réaliser l'opération de codage parcimonieux après échantillonnage, il faut coupler au capteur un convertisseur analogique numérique et un système de calcul, tel qu'un microcontrôleur (programmation en langage C ou assembleur) ou un circuit intégré de type FPGA (Field-programmable Gate Array ; programmation en langage VHDL). C'est ce système qui est chargé d'effectuer les opérations de projection matricielle qui sont, en pratique, simplifiées par le choix d'une matrice d'observation binaire : si le signal numérique est codé en binaire signé, l'opération de changement de signe est extrêmement simple puisqu'il s'agit simplement d'un changement de signe sur le bit adéquat. La figure 5.2 décrit le principe de l'approche numérique.

5.1.1 Choix matériel

Le choix du matériel s'est porté sur un microcontrôleur PIC (Peripherial Interface Controller) de la série PIC 18FXX de Microchip : ces microcontrôleurs présentent des avantages du point de vue pratique, notamment la présence d'un convertisseur analogique-numérique 8/10 bits intégrés, et des fréquences d'horloge suffisantes (48Mhz) pour un nombre d'opérations par seconde atteignant les 12 MIPS. Ce microcontrôleur exécute les

FIGURE 5.3 – PIC 18F4550 implémentation 40 broches .

opérations sur 8 bits. La programmation en langage C est possible, en plus de l'assembleur, dans le but d'assurer une évolutivité. Ce microcontrôleur intègre une mémoire flash de 32 Ko dans laquelle sera stockée la matrice d'observation Φ, et 2 Ko de RAM qui contiendront l'observation y. Le coût est relativement faible – autour de 4€– et la consommation maximale (horloge à 48 MHz, température $+85^0$) est de 50 mA pour une tension de VDD=5V en mode d'exécution et 15μA en mode veille.

5.1.2 Algorithme du microcontrôleur (PIC)

La programmation du microcontrôleur peut s'envisager de diverses manières. L'objectif est d'avoir un algorithme qui s'implémente facilement, flexible en terme de variation des

2. Un tel choix rend aussi la solution analogique plus simple, puisqu'il n'y a plus de multiplication à réaliser.

paramètres (nombre de mesures m, taille de la fenêtre d'observation N) et surtout qui s'exécute rapidement pour pouvoir fonctionner en temps réel.

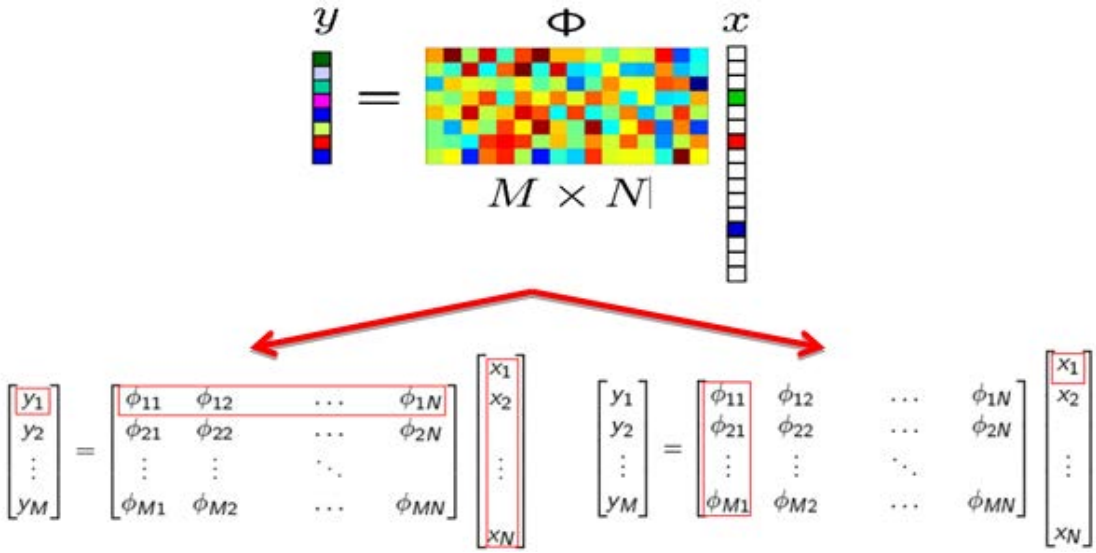

FIGURE 5.4 – Projection de x sur la matrice d'observation Φ. À gauche, on procède suivant les lignes de la matrice, en effectuant chaque produit scalaire entre la ligne de matrice et le vecteur ; à droite, on procède selon les colonnes, en réalisant la combinaison linéaire de colonnes de la matrice d'observation.

L'algorithme doit collecter les échantillons du signal issus du capteur et projeter ceux-ci sur la matrice d'observation Φ avant de transmettre le résultat de cette projection. L'opération matricielle $y = \Phi x$ peut être envisagée de deux manières, comme l'illustre la figure 5.4 : on peut soit mémoriser les N échantillons de x dans le PIC, les opérations s'effectuent selon les lignes de la matrice ; soit traiter les échantillons un à un après chaque acquisition du CAN, les opérations s'effectuent alors selon les colonnes de la matrice (on crée la combinaison linéaire de colonnes). C'est deuxième solution que l'on retient ici. Elle présente l'avantage de ne pas dépendre de la capacité de stockage des N échantillons du signal et de mieux répartir les calculs dans le temps. Cependant, elle reste limitée, dans une moindre mesure, par la vitesse d'exécution des instructions : il faut effectuer les m changement de signes et m additions binaires avant l'arrivée de l'échantillon suivant.

5.1.2.1 Implémentation de la matrice d'observation

La matrice d'observation est mémorisée dans le système par le programme. Elle est fixée au préalable et ne sera pas modifiée.

La figure 5.6 décrit le processus de mémorisation de la matrice : à partir d'une matrice binaire (± 1) générée extérieurement (avec Matlab, par exemple,) on convertit celle-ci en

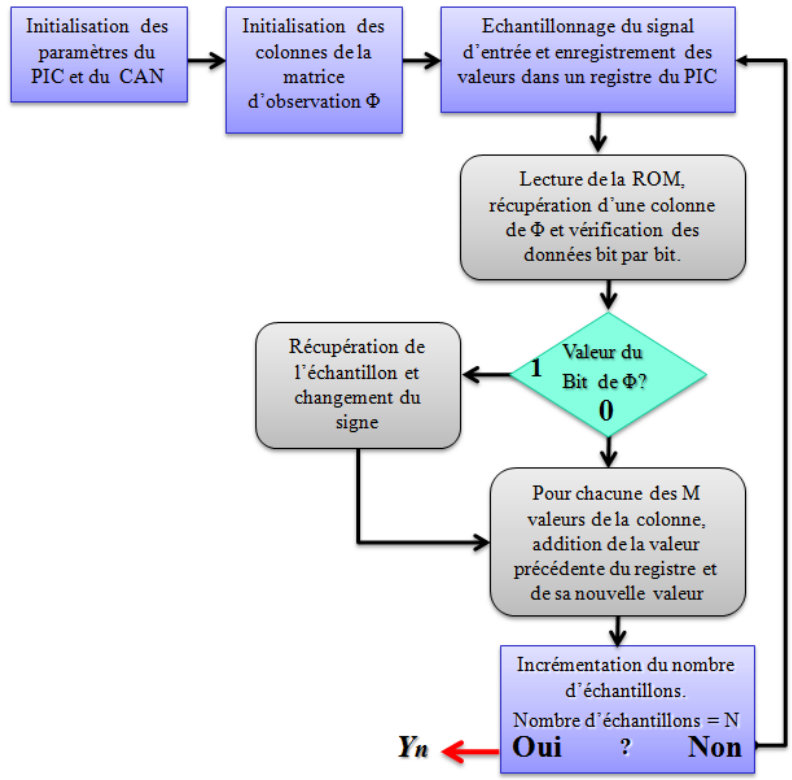

FIGURE 5.5 – Algorithme d'un codeur parcimonieux numérique

binaire selon la règle $\begin{cases} -1 & \Rightarrow & 1 \\ +1 & \Rightarrow & 0 \end{cases}$.[3] Enfin, elle est transposée pour que les colonnes soient stockées en ligne : comme on a fait le choix de traiter le signal échantillon par échantillon, et donc d'utiliser la matrice colonne par colonne, il est plus pratique d'aligner les valeurs des colonnes dans la mémoire.

5.1.3 Mise en œuvre

Ce système est limité par divers facteurs, dont la mémoire RAM et la taille des registres. Le PIC gère les valeurs de l'acquisition y_i sur 16 bits en binaire signé : 65536 valeurs, de -32768 à +32767 et le microcontrôleur traite des valeurs entières du signal $x(t)$ issues du capteur et converties par un CAN 8 bits. Le signal ainsi numérisé X_n est réparti sur 256 valeurs. Le signal étant codé en binaire signé, la valeur absolue maximale d'un échantillon est 128. Dans le pire cas, on aura donc $N_{\max} = \frac{2^{15}}{2^7} = 256$ échantillons utilisables avant qu'il y ait un dépassement d'entier. En pratique, la meilleure façon de se protéger d'un tel souci est de limiter le nombre d'échantillons, ce qui est notre cas puisque nous n'envisageons pas de valeur supérieure à 256. Cependant, les séquences de projections aléatoires ont une valeur

3. Ce choix est purement arbitraire et correspond à un choix fait dans l'implémentation logicielle : si le bit est à 1, alors il y a un changement de signe. Ce choix permet d'implémenter le changement de signe à l'aide d'un opérateur XOR entre le bit de la matrice et le bit de signe de l'échantillon.

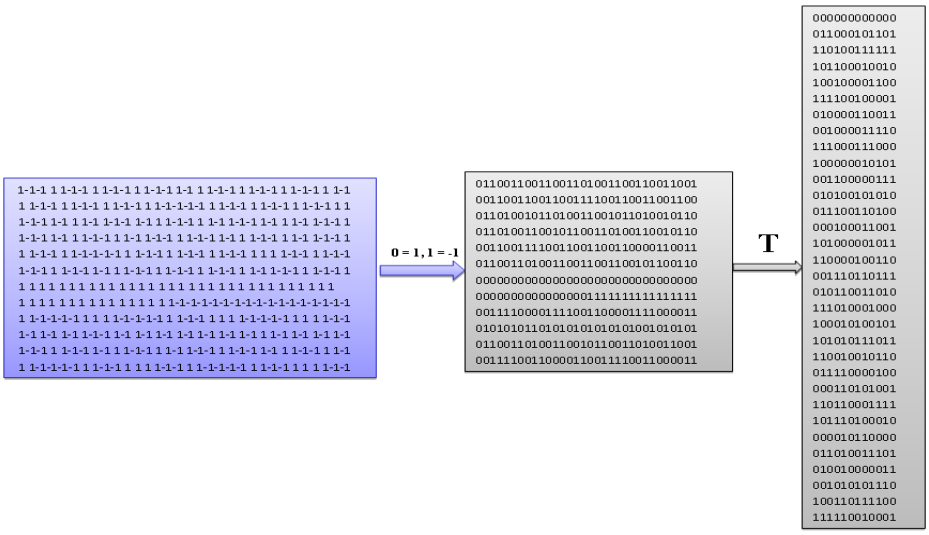

FIGURE 5.6 – *Exemple d'une matrice aléatoire de type Walsh* 12×32 *rangée en fonction des colonnes dans la mémoire flash(32Ko) du PIC.*

moyenne nulle et il y a peu de chance qu'un problème de dépassement d'entier survienne, même si un nombre supérieur d'échantillons est considéré. On peut donc envisager des valeurs de N supérieures à ce $N_{\max}$ sans risque.

Un second problème de dimensionnement est celui du temps de calcul, et du temps de conversion du CAN : le nombre maximal de projections m et la fréquence d'échantillonnage sont liés et limités par le temps de conversion et de mise à disposition de l'échantillon, et le temps de calcul pour effectuer la projection : il faut que cela soit plus court que la période entre deux échantillons. Le temps de calcul exact dépend de la manière dont est programmé le système et sa gestion des mémoires.

5.1.4 Comparaison avec un échantillonneur simple

Pour finir, effectuons une comparaison avec un simple échantillonnage, sans compression.

Considérons le signal d'un accéléromètre, qui sera échantillonné à $F_e = 6.4Khz$ avec le dispositif de la figure 5.7. Le **AD7823** est un CAN usuel de 8 bits (pour obtenir la même résolution que le CAN du PIC18F4550.) La consommation maximale est de $3mW$ pour une tension de $5.5V$ et une fréquence d'échantillonnage d'une dizaine de kilohertz. De plus, pour l'adaptation des entrées, on utilise l'AOP AD8021 qui consomme $35mW$. La consommation totale est donc proche de 40 mW. Si on échantillonne pendant une seconde à $F_e = 6,4$ kHz, on produit alors 51.2 kb de données.

Dans le cas du codeur numérique parcimonieux utilisant un PIC 18F4550, avec une fréquence d'échantillonnage $F_e = 6.4$ kHz, avec $m = 30$ et $N = 256$, la projection du codage

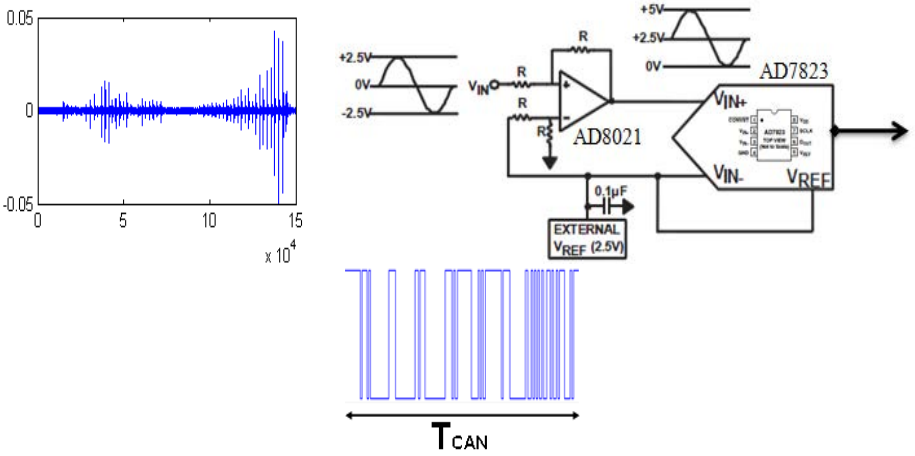

FIGURE 5.7 – *Échantillonnage classique d'un signal provenant des accéléromètres.*

parcimonieux permet d'obtenir 750 échantillons (en 25 groupes de 30), ce qui correspond à 6 kb si les échantillons sont codés sur 8 bits et à 12 kb de données dans notre cas. La taille des données est donc réduite de $76,6\%$.

Cependant, la consommation maximale du microcontrôleur PIC18F4550 est de 190 mW, soit environ 150 mW de plus que le système associant un CAN et un AOP, soit une augmentation de 375% (dans le pire des cas) de la consommation énergétique : il sera donc nécessaire de comparer ceci au gain obtenu au niveau de la transmission.

On retiendra comme limitations la mémoire du microcontrôleur, et la vitesse de calcul qui restreint le nombre de projections : entre chaque acquisition, il faut effectuer les m changements de signe [4] et additions binaires avant l'arrivée de l'échantillon suivant. Cette limite dépend donc de la fréquence d'échantillonnage en entrée, de la fréquence de calcul du microcontrôleur et du nombre m de projections. Il faut en outre être capable d'envoyer le résultat vers la sortie lorsque N échantillons sont arrivés.

5.2 Solution retenue

Compte tenu de l'objectif, à savoir réduire le débit de données tout en ayant un système simple et peu coûteux (en prix et en énergie,) la solution retenue est celle du *codeur parcimonieux numérique* : un système d'échantillonnage traditionnel couplé à un microcontrôleur effectuant la projection sur la matrice d'observation. Ce système est présenté en figure 5.8. Les solutions techniques envisagées dans ce chapitre ont des limites, qui sont principalement dues au choix du PIC. Cependant, des travaux ultérieurs sur un circuit dédié devraient permettre d'avoir un système efficace.

4. En réalité, seulement $m/2$ sont nécessaires en moyenne, mais il faut envisager le pire des cas.

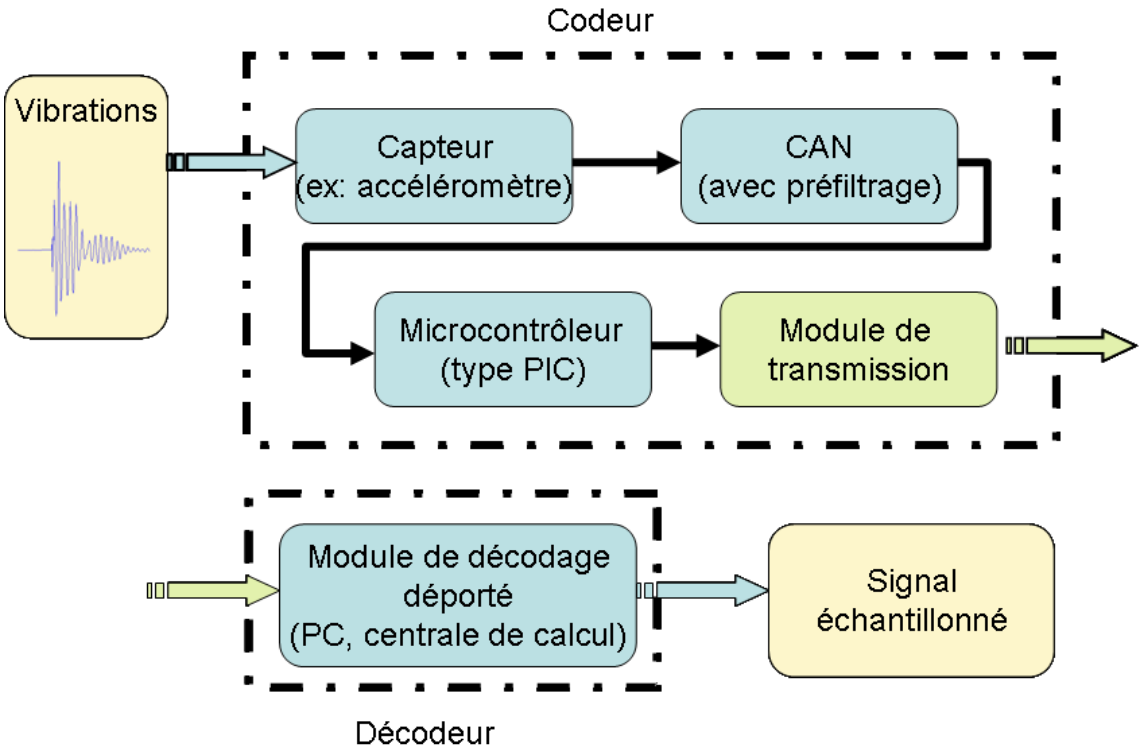

FIGURE 5.8 – Schéma du système retenu.

5.3 Perspectives

Les résultats présentés dans ce chapitre sont le fruit de l'encadrement d'un travail de Master [Kou10]. Ces résultats, ainsi que ceux du chapitre 7 [CHM+12, CHDM12], ont engendré un nouveau projet de Master, ayant pour objectif le développement d'un système de codage sur composant silicium dont les résultats sont très encourageants [CL12] : le coût du système de compression est inférieur au gain obtenu à la transmission des données, ce qui permet de réduire le bilan énergétique final.

Chapitre 6

Apprentissage de dictionnaire

Résumé : *Ce chapitre s'intéresse à l'apprentissage d'un dictionnaire permettant la décomposition parcimonieuse du signal. On y présente deux méthodes,* Focuss-CNDL *et k-SVD, que l'on étudie sur des exemples simples puis sur des signaux synthétiques d'enregistrement neuronal. Ces deux méthodes permettent d'effectuer conjointement l'apprentissage du dictionnaire et celui des coefficients de représentation parcimonieuse des signaux.*

Connaitre un dictionnaire permettant la décomposition parcimonieuse d'un signal est indispensable à la reconstruction d'un signal compressé par une approche de codage parcimonieux. De nombreux exemples de la littérature sur le codage parcimonieux utilisent, comme cela est vu dans les chapitres précédents, les bases usuelles de représentation comme la base canonique, la base de Fourier discrète ou des bases d'ondelettes discrètes. Cependant, tous les signaux ne se décomposent pas nécessairement de manière parcimonieuse sur l'une de ces bases : il s'agit alors de construire (d'apprendre) un dictionnaire qui permettra une telle décomposition. On s'intéresse ici à l'apprentissage à partir d'une banque de données représentative des signaux observés, à partir de laquelle il faut estimer un dictionnaire et donc une représentation parcimonieuse de chaque signal. Ce problème est donc celui de l'estimation conjointe des sources parcimonieuses et du dictionnaire de décomposition parcimonieuse.

On nomme dans ce chapitre $x \in \mathbb{R}^N$ le signal, Ψ de dimension $N \times L$ le dictionnaire et $\alpha \in \mathbb{R}^L$ la source parcimonieuse correspondant à x, l'objectif étant de minimiser $\|x - \Psi\alpha\|_2$. L'estimation conjointe du dictionnaire et des sources repose sur l'observation d'un nombre N_S de signaux. Nous pouvons regrouper ces couples (signal, source) sous forme matricielle, l'ensemble des signaux est regroupé dans la matrice X de taille $N \times N_S$, dont les colonnes s'appellent X_j, et les sources dans la matrice A de taille $L \times N_S$, dont les colonnes s'appellent A_j, cela donne alors $X = \Psi A$.

6.1 Apprentissage hors ligne d'un dictionnaire de décomposition parcimonieuse

L'apprentissage d'un dictionnaire pour des méthodes de codage parcimonieux [O+96] passe par la minimisation d'une fonction de coût qui peut s'écrire sous la forme

$$\begin{aligned} F(X, A, \Psi) &= \sum_{j=1}^{N_S} \|X_j - \Psi A_j\|_2^2 + \lambda \sum_{j=1}^{N_S} P(A_j) \\ &= \|X - \Psi A\|_F^2 + \lambda \sum_{j=1}^{N_S} P(A_j), \end{aligned} \tag{6.1}$$

où $P(A_j)$ est une fonction de pénalité sur la parcimonie du vecteur source (en pratique, on utilise la norme ℓ_p, $0 < p \leq 1$, cf. p. 17), $\|.\|_F$ est la norme de Frobenius et le paramètre λ permet de régler le compromis entre l'erreur de reconstruction et la parcimonie de la source. Plus λ est grand, plus la mesure de parcimonie est pénalisée : cela favorise des sources parcimonieuses. Inversement, quand λ est faible, on favorise l'adéquation entre les sources et les signaux, aux dépens de la parcimonie des sources.

On cherche donc à minimiser l'erreur de reconstruction en norme ℓ_2 (matérialisée par la norme de Frobenius de la matrice d'erreur [1]) tout en maintenant une contrainte sur la parcimonie des sources. L'approche classique [Mal08, AEB06] est de procéder de manière itérative, en alternant l'estimation des sources A_j en fixant le dictionnaire, et l'estimation du dictionnaire Ψ en fixant les sources A. On ne s'attarde pas dans ce chapitre sur l'estimation des sources A, puisqu'il s'agit uniquement de la recherche d'une solution parcimonieuse, qui a été traitée au chapitre 2.2.2.

6.1.1 Approche de Kreutz-Delgado [KDMR+03]

Lorsque l'on cherche à optimiser conjointement les sources et le dictionnaire (éq. (6.1)), on est confronté à un problème de normalisation. En effet, on remarque que pour toute solution (Ψ, A), $(\nu\Psi, \frac{1}{\nu}A)$ est aussi une solution : il est donc tentant de donner des petits coefficients aux signaux (ce qui a pour effet de réduire la norme ℓ_p) et de compenser ceci par des colonnes du dictionnaire à très forte norme. Pour éviter ce phénomène, il faut ajouter une contrainte supplémentaire sur le dictionnaire appris. On peut par exemple contraindre la matrice du dictionnaire à avoir une norme de Frobenius unité, mais cela n'est pas toujours suffisant dans le cas de dictionnaires sur-complets [KDMR+03] : certaines colonnes peuvent devenir inutilisées car leur faible norme entraine des grandes valeurs pour les coefficients correspondants des sources, et donc une forte pénalisation. Il apparaît donc que la méthode proposée par [KDMR+03] consistant à normaliser les colonnes du dictionnaire est plus

1. On remarque que $\|A\|_F^2 = \sum_j \|A_j\|_2^2$.

adaptée. On remarque que cela implique une norme de Frobenius unité si les colonnes sont normalisées à $\|\Psi_j\|_2 = \frac{1}{\sqrt{L}}$, mais la réciproque n'est pas vraie.

L'approche présentée par [KDMR+03] s'appuie sur l'algorithme FOCUSS (cf p. 2.2.2.3) [GR97, GGR95] pour l'estimation des sources parcimonieuses. On rappelle que l'objectif de cet algorithme itératif est de résoudre le problème :

$$(P_p) : \qquad \hat{x} = \arg\min_{A_j} \|X_j - \Psi A_j\|_2 + \lambda \|A_j\|_p.$$

L'itération de mise à jour des sources est adaptative : à dictionnaire Ψ fixé, et à partir de l'estimation courante de la source A_j, cette dernière est améliorée à l'aide de l'équation suivante :

$$\hat{A_j} \leftarrow \Pi^{-1}(A_j)\Psi^T[\Psi\Pi^{-1}(A_j)\Psi^T + \lambda\|A_j\|_p^{1-p} I]^{-1} X_j, \tag{6.2}$$

avec $\Pi^{-1}(A_j) = \text{diag}\left(|A_j(i)|^{2-p}\right)$ et λ le paramètre de régularisation, favorisant la parcimonie de la source lorsqu'il est grand.

L'algorithme proposé procède de manière séquentielle, en mettant à jour les sources à dictionnaire fixé, puis en faisant évoluer le dictionnaire en supposant que les sources sont connues. On s'intéresse donc à la mise à jour du dictionnaire Ψ à partir de l'ensemble des observations $X = [X_1, .., X_{N_S}]$. Cela se traduit par la minimisation de l'erreur quadratique :

$$\hat{\Psi} = \arg\min_{\Psi} \frac{1}{N_S} \sum_{j=1}^{N_S} ||\Psi \hat{A}_j - X_j||_2^2, \tag{6.3}$$

dont la solution est [KDMR+03]

$$\hat{\Psi} = \Sigma_{X\hat{A}} {\Sigma_{\hat{A}\hat{A}}}^{-1}, \tag{6.4}$$

où

$$\Sigma_{X\hat{A}} = \frac{1}{N_S} \sum_{j=1}^{N_S} X_j \hat{A}_j^T \ ; \ \Sigma_{\hat{A}\hat{A}} = \frac{1}{N_S} \sum_{j=1}^{N_S} \hat{A}_j \hat{A}_j^T .$$

Cependant, ce calcul requiert l'inversion d'une matrice de taille $L \times L$, si tant est que celle-ci est inversible. Une alternative consiste à utiliser l'algorithme du gradient descendant [KDMR+03], qui permet d'obtenir itérativement le dictionnaire mis à jour $\hat{\Psi}$ à partir du dictionnaire estimé précédent Ψ :

$$\hat{\Psi} \leftarrow \Psi - \gamma \left(\Psi \Sigma_{\hat{A}\hat{A}} - \Sigma_{X\hat{A}}\right), \tag{6.5}$$

où γ est le pas d'adaptation du gradient. Par la suite, on notera $\Psi\Sigma_{\hat{A}\hat{A}} - \Sigma_{X\hat{A}} = \delta\Psi$. Cette méthode de mise à jour correspond à ce qui était proposé par [OF97], mais ne prend pas en compte la normalisation nécessaire de la matrice Ψ.

Pour ce faire, on procède colonne par colonne :

$$\hat{\Psi}_l = \Psi_l - \gamma \left(I_N - \frac{\Psi_l \Psi_l^T}{\|\Psi_l\|_2^2} \right) \delta\Psi_l, \tag{6.6}$$

pour l allant de 1 à L et où $\delta\Psi_l$ représente la l-ème colonne de $\delta\Psi$ définie plus haut. Les colonnes sont ensuite normalisées :

$$\hat{\Psi}_l \leftarrow \frac{\hat{\Psi}_l}{\sqrt{L}\|\hat{\Psi}_l\|_2}.$$

Les auteurs de [KDMR+03] appellent cet algorithme d'apprentissage de dictionnaire avec colonnes normalisées : FOCUSS-CNDL FOCUSS-CNDL.

6.1.2 Approche K-SVD [AEB06]

Cette méthode cherche à trouver le meilleur dictionnaire possible pour une représentation parcimonieuse, qui réponde à l'objectif

$$\min_{\Psi, A} \left\{ \|X - \Psi A\|_F^2 \right\} \text{ tel que } \forall j,\ \|A_j\|_0 \leq T_0, \tag{6.7}$$

où l'entier T_0 est fixé de manière arbitraire, $T_0 << N$. Le principe reste identique à ce qui a été discuté précédemment, à savoir procéder de manière itérative en recherchant tour à tour le dictionnaire le plus adapté et la décomposition la plus parcimonieuse possible. C'est sur l'étape de mise à jour du dictionnaire qu'il y a une distinction, la recherche de solution parcimonieuse étant possible avec diverses méthodes, dont certaines ont été présentées au chapitre 2.2.2. Les auteurs de [AEB06] font le choix d'utiliser l'algorithme OMP, qui permet d'obtenir facilement une solution parcimonieuse avec un nombre maximal de composantes fixé par avance.

Le processus de mise à jour du dictionnaire traite chaque colonne une à une : on décompose le produit ΨA de (6.7) en une somme de L matrices de rang 1 : $\Psi A = \sum_{i=1}^{L} \Psi_i A^i$, où A^i représente la i-ème ligne de la matrice A. Parmi ces éléments, on isole le terme $\Psi_l A^l$ dans le but de le mettre à jour en figeant les autres : on s'intéresse uniquement à la l-ème colonne Ψ_l de Ψ et la l-ème ligne de la matrice A. Le critère devient alors la minimisation

selon Ψ_l et A^l de :

$$\begin{aligned} \|X - \Psi A\|_F^2 &= \left\| X - \sum_{i=1}^{L} \Psi_i A^i \right\|_F^2 \\ &= \left\| X - \left(\sum_{i \neq l} \Psi_i A^i \right) - \Psi_l A^l \right\|_F^2 \\ &= \left\| E^l - \Psi_l A^l \right\|_F^2, \end{aligned} \tag{6.8}$$

en notant $E^l = X - \left(\sum_{i \neq l} \Psi_i A^i\right)$ la matrice de taille $N \times N_S$ représentant l'erreur sur les N_S signaux lorsqu'on ne tient plus compte de la contribution de la l-ème colonne Ψ_l du dictionnaire Ψ. La décomposition en valeurs singulières (SVD) permet d'obtenir la solution minimisant (6.8) [EY36] : la SVD décompose E^l sous la forme $U\Delta V^T$, où U et V sont des matrices unitaires (dont les colonnes sont orthonormées) et Δ une matrice diagonale dont les éléments sont des réels positifs ou nuls, rangés par ordre décroissant, appelés valeurs singulières de E^l. [EY36] montre que la SVD permet d'obtenir la matrice de rang r donnant la meilleure approximation (au sens de la norme de Frobenius) d'une matrice donnée, sous la forme $U\tilde{\Delta}V^T$ où $\tilde{\Delta}$ ne compte que les r plus grandes valeurs singulières, les autres étant mises à 0. Puisqu'on cherche une matrice de rang 1, on définit alors $\hat{\Psi}_l$ comme la première colonne de U, et $\hat{A}^l$ comme la première colonne de V multipliée par le premier élément de Δ (c'est-à-dire la plus grande valeur singulière). Cependant, cette opération n'impose aucune parcimonie à la nouvelle ligne de coefficients $\hat{A}^l$. Pour remédier à ce problème, [AEB06] propose de ne s'intéresser qu'aux signaux qui utilisent la colonne Ψ_l du dictionnaire. On définit alors l'ensemble ω_l de la manière suivante :

$$\omega_l = \left\{ i | 1 \leq i \leq M,\ A^l(i) \neq 0 \right\}. \tag{6.9}$$

On note $|\omega_l|$ le cardinal de l'ensemble ω_l. On définit aussi la matrice Ω_l de taille $N_S \times |\omega_l|$, valant 1 pour les éléments $(\omega_l(i), i)$ et zéro ailleurs. $A_R^l = A^l\Omega_l$ est alors un vecteur de longueur $|\omega_l|$ ne contenant que les éléments non nuls de A^l. De manière identique, $X\Omega_l$ est une matrice de taille $N \times |\omega_l|$ ne contenant que les signaux dont la décomposition parcimonieuse utilise le l-ème atome Ψ_l du dictionnaire Ψ ; et $E_R^l = E^l\Omega_l$ la représentation des erreurs correspondant à ces signaux lorsque la contribution de la colonne Ψ_l est retirée.

L'objectif est de minimiser (6.8) selon Ψ_l et A^l, mais en forçant la solution $\tilde{A}^l$ à avoir le même support que A^l. Cela revient à minimiser

$$\left\| E^l\Omega_l - \Psi_l A^l\Omega_l \right\|_F^2 = \left\| E_R^l - \Psi_l A_R^l \right\|_F^2. \tag{6.10}$$

Cette fois, on peut directement appliquer la SVD à E_R^l, qui la décompose en $U\Delta V^T$. On définit alors $\tilde{\Psi}_l$ comme la première colonne de U, et $\tilde{A_R^l}$ comme la première colonne de V

multipliée par $\Delta(1,1)$ [EY36]. De cette manière, les colonnes de Ψ restent normalisées car les colonnes de U sont orthonormées, et le support de chaque représentation reste le même (ou se réduit) puisqu'on n'a considéré que les colonnes présentant des éléments non nuls au départ : les éléments déjà nuls n'ont pas été modifiés. De plus, comme on met à jour les sources en même temps que chaque colonne du dictionnaire, ceci accélère la convergence de l'algorithme.

6.2 Simulations sur des exemples simples

6.2.1 Description des méthodes

Les méthodes d'apprentissage de dictionnaire présentées dans la section précédente proposent une solution pour la construction de dictionnaires adaptés à la décomposition parcimonieuse des signaux. Une première étape de validation de ces méthodes porte sur les simulations sur données synthétiques, afin d'évaluer l'implémentation et l'influence des divers paramètres.

Pour rappel, on utilisera les notations suivantes :

- N_S : nombre de signaux considérés
- $A = [A_1, ..., A_{N_S}]$: l'ensemble des signaux "sources" $A_j \in R^L$ parcimonieux. A_j est la j-ème source ($j = 1, ..., N_S$).
- $X = [X_1, .., X_{N_S}]$: l'ensemble des signaux observables dont les éléments de A représentent la décomposition (parcimonieuse) dans Ψ : $X = \Psi A$. $X_j \in R^N$ est la j-ème observation ($j = 1, ..., N_S$).
- $\Psi = [\Psi_1, ..., \Psi_L]$: le dictionnaire de dimension $N \times L$ de décomposition parcimonieuse. $\Psi_l \in R^N$: la l-ème colonne de Ψ ($l = 1, ..., L$)

Lors des simulations, nous avons procédé comme suit pour initialiser le problème, quel que soit l'algorithme testé :

1. Une matrice Ψ^0 est générée de manière aléatoire, distribution normale $\mathcal{N}(0,1)$;
2. Des sources parcimonieuses A^0 sont générées aléatoirement avec un nombre fixé k de composantes non nulles ;
3. Les signaux observables X sont définis à partir de Ψ^0 et A^0 : $X = \Psi^0 A^0$;
4. Une première approximation $\tilde{\Psi}$ du dictionnaire estimé est obtenue à partir des L premières colonnes de X ;

Ensuite, nous appliquons l'algorithme considéré.

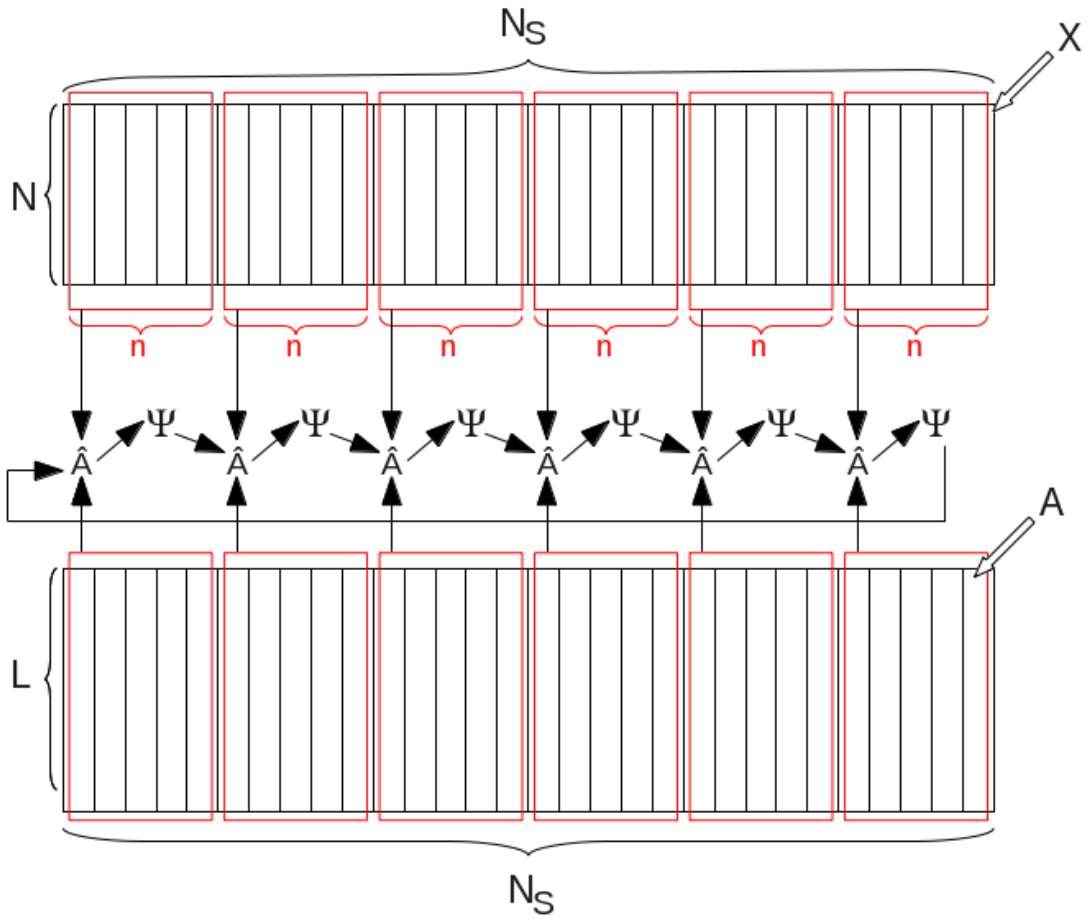

FIGURE 6.1 – Pour appliquer FOCUSS-CNDL, on subdivise l'ensemble des N_S observations en sous-ensembles de taille n avec $n < N_S$. Pour chaque sous-ensemble, on met à jour les n sources à l'aide de FOCUSS (éq. (6.2)) et de l'estimation courante du dictionnaire Ψ. Ensuite, le dictionnaire est mis à jour (éq. (6.6)) puis on recommence avec le bloc de taille n suivant jusqu'à avoir parcouru l'ensemble des N_S signaux. L'opération est itérée jusqu'à convergence ou un critère d'arrêt arbitraire.

6.2.1.1 Mise en œuvre des algorithmes

L'algorithme K-SVD est implanté comme décrit dans l'article [AEB06] (cf. B) et ceci ne requiert pas d'explication au delà de la présentation faite précédemment.

La mise en œuvre de FOCUSS-CNDL s'appuie sur la description faite par [KDMR+03]. Après l'initialisation décrite précédemment (choix des L premiers signaux pour initialiser le dictionnaire puis pseudo-inverse de Moore-Penrose pour initialiser les sources), on procède à l'estimation de sources parcimonieuses à l'aide d'une itération de FOCUSS (éq. (6.2)) pour les n premiers signaux ($n < N_S$, et diviseur de N_S pour plus de facilité). Une fois ces n estimations faites, le dictionnaire est mis à jour selon l'équation (6.6), et on recommence jusqu'à avoir traité tous les signaux de X. Cette méthode est représentée par la figure 6.1. Cette opération est répétée un certain nombre de fois ([KDMR+03] le fait 200 fois).

Cependant, surtout lors des premières itérations, une seule itération de FOCUSS ne conduit généralement pas à un vecteur parcimonieux. Pour accélérer la convergence, le calcul des termes $\Sigma_{X\hat{A}}$ et $\Sigma_{\hat{A}\hat{A}}$ est effectué à partir de sources rendues parcimonieuses en ne retenant que les T_0 coefficients de plus grande amplitude. La valeur de T_0 est arbitrairement réglée à k, puisque celui-ci est connu.

6.2.1.2 Performances de l'algorithme

L'objectif de l'algorithme est d'apprendre un dictionnaire permettant d'obtenir une décomposition parcimonieuse de nos signaux. Il y a donc deux critères à respecter :

- les sources estimées $\hat{A}$ doivent être parcimonieuses ;
- l'erreur $\|X - \Psi A\|_F$ doit être minimale.

Dans le cas de la simulation, puisque le dictionnaire utilisé lors de la création des vecteurs-signaux est connu, il est possible de s'intéresser aussi à la correspondance entre le dictionnaire d'origine et le dictionnaire appris. Ci-après, nous discutons les possibilités de mesure de l'erreur d'estimation du dictionnaire, puis celles de l'estimation des sources.

6.2.1.3 Mesure de l'erreur d'estimation du dictionnaire

Pour mesurer la qualité d'estimation du dictionnaire, puisque l'on connait le dictionnaire original, une idée intuitive est de simplement mesurer $\left\|\hat{\Psi} - \Psi\right\|_F$. Cependant, il n'y a rien dans la méthode qui contraint le dictionnaire appris à présenter les colonnes dans un ordre identique à celui du dictionnaire original : il est donc opportun d'ordonner arbitrairement les colonnes dès le départ, par exemple par ordre croissant des coefficients de la première ligne de la matrice. Toutefois, ceci n'est pas suffisant : comme on l'a déjà mentionné, pour toute solution (Ψ, A), $(\nu\Psi, \frac{1}{\nu}A)$ est aussi une solution. Pour cela, on a choisi une normalisation des colonnes du dictionnaire, cependant cela laisse encore un degré de liberté ($\nu = \pm 1$). On peut alors aussi imposer que les coefficients de la première ligne soient tous positifs. Ajouter ces contraintes (positivité de la première ligne et ordre croissant) permet de réordonner le dictionnaire à l'identique à condition que la reconstruction soit exacte : si ce n'est pas le cas, il est aisé d'avoir une erreur si les éléments de la première ligne sont proches les uns des autres, ou proches de zéro (provoquant un changement de signe sur l'ensemble de la colonne parce que le premier élément est estimé négatif au lieu d'être positif). Dans la pratique, ces contraintes rendent difficile l'exploitation de cette mesure.

Nous avons aussi envisagé d'utiliser une mesure d'angle pour qualifier l'estimation. Les dictionnaires appris peuvent aussi être considérés comme des opérateurs de projection de $\mathbb{R}^L$ sur un sous-espace de dimension maximale N. On pourrait alors comparer l'angle ([BG73], cf. p. 121) entre les deux sous-espaces engendrés par les lignes du dictionnaire d'origine et celles du dictionnaire appris. Si celui-ci est nul, c'est que les sous-espaces sont

superposés, les dictionnaires pourraient alors être identiques. Cependant, il s'avère qu'une telle méthode est très sensible à la permutation de deux colonnes dans le dictionnaire, puisque cela change complètement le sous-espace représenté par les lignes du dictionnaire. On retrouve le problème précédent lié à l'ordonnancement des colonnes du dictionnaire.

L'approche qui nous semble donc la plus efficace est la mesure de corrélation entre les colonnes du dictionnaire appris et celui d'origine. Dans l'idéal, si le dictionnaire estimé correspond au dictionnaire original (au sens où il permet de décrire de manière aussi parcimonieuse les sources), chacune de ses colonnes doit être fortement corrélée avec une seule colonne du dictionnaire d'origine. Cela suppose que les colonnes de celui-ci sont faiblement intercorrélées (on retrouve l'idée de cohérence du dictionnaire mentionnée p. 39). Nous nous intéressons donc au nombre de colonnes du dictionnaire d'origine ayant une colonne estimée correspondante, c'est-à-dire avec une corrélation supérieure à 0,95.

6.2.1.4 Mesure de l'erreur sur l'estimation des sources parcimonieuses

Pour mesurer l'erreur sur les sources estimées, on peut procéder de manière classique, en calculant l'erreur au sens de la norme ℓ_2 entre les sources estimées et celles simulées. Cela suppose que le dictionnaire estimé est correct d'une part, et d'autre part que ses colonnes sont ordonnées de la même manière que celles du dictionnaire original : c'est toujours le même problème. Si ce n'est pas le cas, cette mesure d'erreur n'a pas de sens. Dans ce cas là, une autre possibilité d'évaluation de la qualité du dictionnaire estimé consiste en la seule mesure de la parcimonie des sources estimées, au travers des méthodes présentées page 17. Notamment, puisque les sources sont été générées comme strictement parcimonieuses, avec le même nombre de composantes, il s'agit de s'assurer qu'il n'y a pas de disparité trop importante du côté parcimonieux des différentes sources estimées.

6.2.2 Résultats et commentaires

L'algorithme FOCUSS-CNDL s'appuie sur les équations (6.2) pour l'estimation des sources parcimonieuses et (6.6) en ce qui concerne la mise à jour des colonnes du dictionnaire. Ces équations sont rappelées ci-dessous :

$$\hat{A}_j \leftarrow \Pi^{-1}(A_j)\Psi^T[\Psi\Pi^{-1}(A_j)\Psi^T + \lambda\|A_j\|_p^{1-p}I]^{-1}x,$$

$$\hat{\Psi}_l \leftarrow \Psi_l - \gamma\left(I_N - \Psi_l\Psi_l^T\right)\delta\Psi_l.$$

Dans ces équations, il y a trois paramètres importants : p, γ et λ. p correspond à la norme ℓ_p que l'itération de FOCUSS cherche à minimiser, γ est le pas d'adaptation du gradient de mise à jour du dictionnaire et λ caractérise le terme de régulation de FOCUSS, qui permet aussi de régler l'importance relative de la norme ℓ_p des sources par rapport à l'erreur

$\|x - \Psi\alpha\|_2$. Ces trois paramètres influent fortement sur les résultats d'apprentissage, si bien qu'il est difficile de répéter les résultats présentés par les auteurs de [KDMR$^+$03].

Le paramètre λ permet de régler le compromis entre l'erreur et la parcimonie des sources estimées. Il peut être réglé au cours des itérations : il est donc intéressant de le choisir très petit au départ afin de ne pas prendre trop en compte la mesure de parcimonie des sources, puisque ces dernières sont mal estimées et pas parcimonieuses, puis de le faire croître jusqu'à une valeur $\lambda_{\texttt{max}}$ lorsque l'estimation devient plus correcte pour donner de l'importance aux sources à ce moment-là. Plusieurs méthodes sont proposées pour régler le paramètre λ par [KDMR$^+$03] (et d'autres plus évoluées sont évoquées, mais non retenues à cause de leur complexité algorithmique, à répéter à chaque estimation d'une source, soit N_S fois par itération) : la première est une simple règle d'évolution croissante monotone fonction du nombre d'itération. Est proposée aussi une méthode se basant sur l'erreur d'estimation, de cette forme :

$$\lambda = \lambda_{\texttt{max}} \left(1 - \frac{\|X_j - \Psi A_j\|_2}{\|X_j\|_2}\right),$$

que l'on utilise ici. La valeur de $\lambda_{\texttt{max}}$ va donc déterminer la qualité de reconstruction et de parcimonie des sources. Si celle-ci est faible, l'erreur $\|X_j - \Psi A_j\|_2$ va être stable quelque soit j. Si $\lambda_{\texttt{max}}$ est plus grand, la variabilité de $\|X_j - \Psi A_j\|_2$ en fonction de j est plus importante, permettant d'obtenir dans quelques cas, malgré une moyenne moins bonne, une erreur plus faible qu'avec un $\lambda_{\texttt{max}}$ plus petit, mais les sources sont en contrepartie plus parcimonieuses.

Le paramètre p influe sur le côté parcimonieux des estimations de signaux. Comme on s'y attend, plus p est petit, plus les estimations sont parcimonieuses. À l'inverse, plus la valeur est proche de 1, plus l'erreur $\|X_j - \Psi A_j\|_2$ est faible.

Le paramètre γ influe sur la vitesse d'évolution du dictionnaire. Si elle est trop faible, l'algorithme n'évoluera pas assez vite, mais si elle est trop élevée, l'estimation du dictionnaire aura du mal à se stabiliser.

En fin de compte, les résultats dépendent fortement de ces paramètres, mais aussi des conditions initiales de la simulation, c'est à dire le dictionnaire et les sources générés aléatoirement : pour un ensemble de paramètres fixés, le bon résultat pour un couple (Φ, A) n'assure pas de bons résultats pour d'autres dictionnaires ou d'autres sources. Inversement, pour un couple (Φ, A) qui permet un bon résultat (dictionnaire appris parfaitement : on arrive à réordonner les colonnes, les colonnes du dictionnaire appris correspondent à celles du dictionnaire de départ, etc), un changement des paramètres, même petit, modifie complètement le résultat. Le choix de la subdivision n des signaux influe aussi sur les résultats (mais cela peut aussi s'expliquer par le fait que lorsque n est plus grand, à nombre d'itérations fixé, le nombre de mises à jour du dictionnaire est plus petit).

En somme, la méthode FOCUSS-CNDL permet l'apprentissage d'un dictionnaire, mais

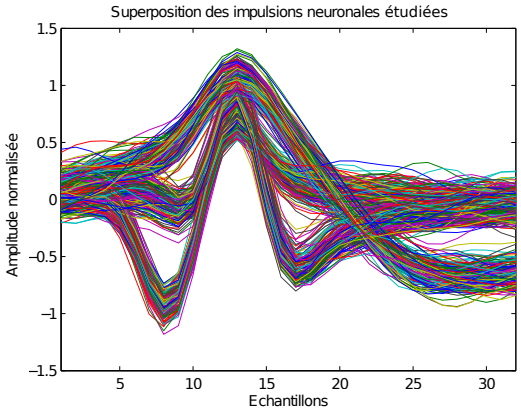

FIGURE 6.2 – Ensemble des 500 signaux synthétiques utilisés pour l'apprentissage.

son paramétrage est très sensible et la mise en œuvre n'est pas évidente. Dans certains cas, aucune colonne estimée n'a de corrélation supérieure à 0,95 avec une colonne du dictionnaire d'origine.

La méthode k-SVD n'a pas autant de paramètres : on ne peut paramétrer que le choix de T_0 limitant le nombre de coefficients des sources calculées au fur et à mesure. Ici, par construction, les sources calculées à chaque étape sont toujours très parcimonieuses. L'algorithme réussit généralement à reconstruire au minimum quelques colonnes du dictionnaire avec une corrélation supérieure à 0,95. Il n'est par contre pas plus évident d'obtenir une reconstruction exacte de toutes les colonnes qu'avec FOCUSS-CNDL.

On pourrait se poser la question de savoir si la qualité de reconstruction d'une colonne dépend de sa fréquence d'utilisation par les sources. Ce n'est apparemment pas le cas car si les colonnes les moins bien représentées font partie de celles qui sont le moins bien reconstruites, la majorité des colonnes mal reconstruites sont autant utilisées que les colonnes ayant une forte corrélation avec une colonne du dictionnaire estimé. Une part importante est même plus utilisée par les sources que la moyenne des colonnes ayant une bonne estimation. D'ailleurs, parmi les colonnes fortement corrélées, il n'apparaît pas de lien entre la valeur de corrélation et le nombre de sources utilisant la colonne. Cependant, même si les colonnes ne sont pas toutes utilisées autant de fois, elles le sont toutes suffisamment et il n'y a pas de colonnes pas ou très peu utilisée. Une fréquence d'utilisation élevée d'un atome du dictionnaire ne conduit donc pas nécessairement à une bonne estimation, pas plus qu'une utilisation relativement plus faible ne conduit à une mauvaise estimation.

6.3 Application de l'apprentissage sur des signaux neuronaux

Nous avons appliqué les méthodes d'apprentissage sur des signaux synthétiques d'impulsions neuronales. La banque de signaux était constituée de $N_S = 500$ signaux de longueur $N = 32$ (cf. description p. 82). Une analyse visuelle des signaux (fig. 6.2) laisse

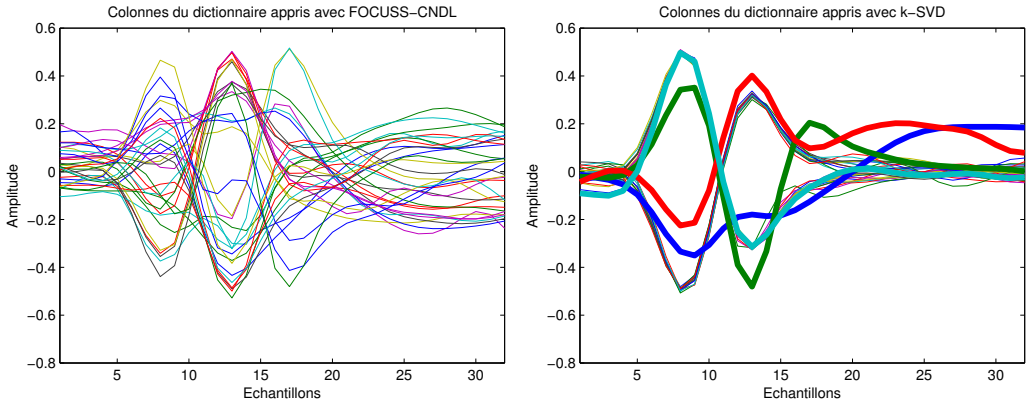

FIGURE 6.3 – Ensemble des colonnes des dictionnaires appris, avec FOCUSS-CNDL à gauche, et k-SVD à droite. Pour ce dernier, les 4 premières colonnes sont mises en évidence par un tracé plus épais.

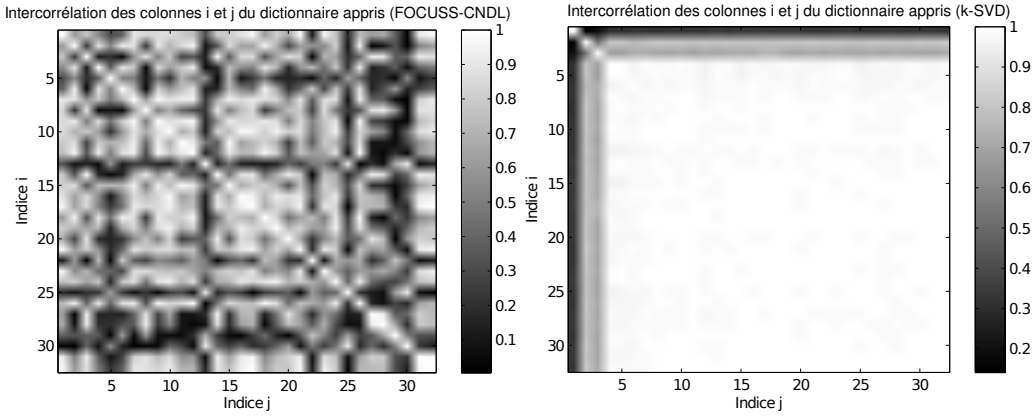

FIGURE 6.4 – Valeur absolue des corrélations entre colonnes du dictionnaire ($\Psi^T\Psi$), avec FOCUSS-CNDL à gauche, et k-SVD à droite.

penser qu'un dictionnaire de décomposition parcimonieuse existe : on y voit en effet apparaître trois groupes ayant chacun une forme de base propre. Ces formes de base pourraient certainement constituer trois atomes du dictionnaire. De plus, une analyse en composantes principales montre que les deux premières composantes suffiraient à décrire en grande partie les signaux (cf. p. 84).

Les dictionnaires appris sont représentés sur la figure 6.3 : ils sont de dimensions 32×32 et sont donc des bases puisque leur rang est bien de 32. On peut constater que les colonnes de ces dictionnaires apparaissent fortement corrélées, ce qui est confirmé par les figures 6.4. Le dictionnaire appris par k-SVD montre bien que 4 colonnes (dont le tracé est plus gras sur la figure 6.3) sont fortement distinctes , alors que les 28 restantes sont fortement corrélées entres elles, ainsi qu'à la quatrième, alors que T_0 a été réglé à 4. Si l'on observe ces 4 formes, elles ne ressemblent pas directement à l'une des trois formes de départ, contrairement à l'intuition formulée au paragraphe précédent. En fin de compte, près de

82% des intercorrélations entre colonnes sont supérieures à 0,95, alors que moins de 7% sont inférieures à 0,5. Les méthodes de tri de colonnes suggérées par [AEB06] prennent leur sens ici : étant donné les fortes valeurs d'intercorrélation entre les 28 dernières colonnes, on peut supposer une forte redondance et donc un apport peu sensible à la qualité du dictionnaire. Cependant, étant donné le manque de diversité des signaux appris, issus de seulement 3 souches, il n'est pas certain qu'il soit possible de changer ce niveau d'intercorrélation. Le dictionnaire appris avec FOCUSS-CNDL ne souffre pas autant de ces problèmes de corrélations des colonnes, puisque moins de 5% des intercorrélations entre colonnes sont supérieures à 0,95, et plus de 40% sont inférieures à 0,5.

On peut s'intéresser à l'erreur moyenne entre X et ΨA, mesurée par $\|X - \Psi A\|_F / N_S$, pour des reconstructions par N_c itérations d'OMP (à distinguer des T_0 itérations utilisées par l'algorithme d'apprentissage) et IRLS (dont le résultat est alors tronqué aux N_c valeurs les plus significatives).

Méthode d'apprentissage	Algorithme	N_c =2	N_c =3	N_c =4	N_c =6	N_c =8
k-SVD, 32×32	OMP	0,20	0,12	0,09	0,05	0,04
k-SVD, 32×40	OMP	0,23	0,12	0,08	0,05	0,04
FOCUSS-CNDL, 32×32	OMP	0,19	0,16	0,13	0,09	0,06
k-SVD, 32×32	IRLS, $p = 0,25$	1,2	1,3	1,6	2	3,16
k-SVD, 32×40	IRLS, $p = 0,25$	1,91	1,8970	1,8247	1,6316	1,4047
FOCUSS-CNDL, 32×32	IRLS, $p = 0,25$	15	20	20	15	14

TABLE 6.1 – Valeur de $\|X - \Psi A\|_F / N_S$, pour les N_c coefficients les plus importants, selon la méthode d'apprentissage et l'algorithme utilisé pour chercher les coefficients sources.

Les résultats sont présentés dans le tableau 6.1. On constate que l'algorithme IRLS ne permet pas une bonne reconstruction : ceci est peut-être causé par la forte corrélation entre les colonnes de la matrice Ψ ou bien la matrice à inverser de l'équation (2.7) de l'algorithme ne s'inverse plus correctement. Les solutions trouvées ne sont alors plus parcimonieuses, d'où l'erreur importante lorsque l'on tronque le résultat en ne gardant que quelques composantes. L'algorithme OMP ne souffre pas de ce problème. Bien que le dictionnaire appris par k-SVD possède apparemment moins de diversité dans ses colonnes (fort taux de corrélation entre elles) que celui appris par FOCUSS-CNDL, les résultats sont légèrement meilleurs, mais cette différence pourrait être expliquée par une différence de vitesse de convergence.

Ces apprentissages ont été effectués sans données de contrôle permettant de vérifier que l'apprentissage n'est pas sur-ajusté. Pour vérifier que ce n'est pas le cas, nous avons effectué un apprentissage avec la méthode k-SVD sur seulement 70% des données, en conservant le reste pour le contrôle. Les résultats sont présentés dans le tableau 6.2 et montrent que dans ce cas, le dictionnaire n'est pas sur-ajusté puisque les résultats sur les données de contrôle et d'apprentissage sont quasiment identiques. De plus, l'écart-type sur l'erreur est

très similaire entre le jeu de contrôle et le jeu d'apprentissage : le dictionnaire appris est très bien ajusté, puisqu'il fonctionne aussi bien pour les deux jeux de données. Ce résultat n'est pas étonnant puisque les signaux sont synthétiques et sont tous générés à partir de l'une des trois impulsions de bases.

	N_c =2	N_c =3	N_c =4	N_c =6	N_c =8
Données d'apprentissage : EQM	0,17	0,11	0,08	0,05	0,03
σ	0,08	0,05	0,03	0,02	0,01
Données de contrôle : EQM	0,18	0,12	0,08	0,05	0,03
σ	0,08	0,05	0,03	0,02	0,01

TABLE 6.2 – Valeur de $\|X - \Psi A\|_F / N_S$ et écart-type σ après N_c itérations d'OMP, pour les données d'apprentissage ($N_d = 350$) et les données de contrôle ($N_d = 150$).

6.4 Conclusion

L'apprentissage de dictionnaire est un problème qui n'est pas simple. Les simulations montrent que les résultats ne sont généralement pas parfaits : il est difficile d'obtenir un dictionnaire estimé correspondant exactement au dictionnaire d'origine. Cependant, les dictionnaires appris sont souvent proches du résultat attendu, les sources estimées peuvent être très parcimonieuses dans l'ensemble, avec quelques cas moins parcimonieux. On retient que l'algorithme FOCUSS-CNDL dispose de nombreux paramètres qu'il conviendra de choisir à bon escient, et dont l'influence sur le résultat n'est pas négligeable. De ce point de vue, l'algorithme k-SVD est beaucoup plus simple, puisqu'on ne choisit que la valeur maximale T_0 du nombre de composantes que l'on veut donner aux sources parcimonieuses. Il faudra cependant veiller à ne pas la choisir trop petite.

Dans une application pratique, ces méthodes permettent toutefois d'apprendre un dictionnaire capable de représenter convenablement et de manière parcimonieuse les signaux traités dans notre exemple, avec une erreur faible lorsqu'on limite le nombre de coefficients de représentation. Il conviendrait d'étendre les simulations à d'autres données, pour généraliser (ou non) ce constat.

De plus, nous nous sommes ici intéressés à seulement deux méthodes parce qu'elles utilisent deux approches intéressantes et différentes, mais pour être exhaustif, il faudrait aussi étudier les autres méthodes existant dans la littérature (comme [MBP+08]).

6.4.1 Améliorations possibles

Les simulations, notamment avec les signaux neuronaux synthétiques, mettent en avant des lacunes dans les algorithmes, qu'il conviendrait de combler pour en améliorer les résultats. On constate que, pour l'algorithme k-SVD, les colonnes du dictionnaire appris

peuvent être très fortement corrélées. Comme le suggère [AEB06], plusieurs améliorations peuvent être envisagées, en travaillant sur deux points : les colonnes très peu utilisées dont on peut supposer qu'elles n'apportent rien ; et les colonnes fortement corrélées entre elles, ce qui laisse entendre une redondance inutile. Dans les deux cas, les colonnes concernées peuvent être remplacées par un terme susceptible de manquer : le signal le moins bien estimé par le dictionnaire (après normalisation) par exemple.

Une autre possibilité d'amélioration est d'avoir un dictionnaire dont la taille peut évoluer. Lors des simulations sur des cas simples, on connaissait le dictionnaire d'origine et sa dimension : on cherchait donc à en obtenir un de même taille. Dans une situation plus réaliste, on ne connait pas cette dimension. L'idée serait alors de lancer l'algorithme avec un grand nombre d'atomes de départ, pour éventuellement retirer des colonnes lorsque celles-ci sont apparemment inutiles. Cela soulève le problème de la dimension nécessaire pour représenter correctement les signaux, et il y aurait éventuellement des questions à se poser avant de diminuer la taille du dictionnaire, comme de savoir si les signaux sont suffisamment bien représentés par les colonnes actuelles ou non, par exemple. Selon les cas, il faut pouvoir retirer une colonne (et donc réduire la taille du dictionnaire) ou remplacer une colonne (dans le but de mieux représenter les signaux) ou bien valider le dictionnaire estimé.

Chapitre 7

Exploitation dans le domaine compressé - Classification de signaux neuronaux

Résumé : *Nous nous intéressons ici aux possibilités d'exploitation du signal compressé en se passant de l'étape de reconstruction. Au travers d'un exemple sur des signaux synthétiques neuronaux, on constate qu'il est possible d'effectuer une classification des signaux à partir de leur forme compressée. Ensuite, nous nous intéressons à la conservation des distances – ce qui justifie la possibilité de classification – par projection sur la matrice d'observation en proposant une probabilité minimale applicable dans un contexte de petites dimensions.*

Dans les chapitres précédents, nous avons vu que la décompression était l'étape coûteuse du codage parcimonieux. Si les méthodes de reconstruction permettent de retrouver le signal à partir de sa forme compressée, c'est parce que l'information portée par le signal se retrouve intégralement dans la forme compressée. On peut donc se demander si cette décompression est indispensable, ou si, à l'inverse, certains traitements peuvent s'appliquer directement aux signaux compressés. Examinons pour cela un exemple de classification, appliqué à des signaux neuronaux.

7.1 Résultats expérimentaux de classification après codage

7.1.1 Contexte expérimental

Les interfaces cerveau-machine (BMI, pour *Brain Machine Interfaces*) visent à établir un lien de communication direct entre le cerveau et des actionneurs extérieurs, tels qu'un curseur informatique ou un instrument robotisé [GNL+08, SMA+08, VPS+08]. Parmi les applications potentielles, il y a la restauration des fonctions sensorimotrices de patients

souffrant de lésion de la moelle épinière, de conséquences d'accidents cardio-vasculaires ou d'autres troubles neurologiques [HSF+06]. Pour cela, un système BMI classique comprend en général quatre éléments : un système d'enregistrement des signaux neurophysiologiques du cerveau, un algorithme d'interprétation qui convertit ces signaux en une variable correspondant à l'action à effectuer, un actionneur, et un retour vers l'utilisateur. Les micro-électrodes implantées dans le cortex permettent la collecte de données relatives à l'activité électrique de groupes neuronaux, ce qui autorise ensuite le contrôle d'appareils extérieurs avec une grande précision. Chaque électrode enregistre l'activité de plusieurs neurones, ce qui implique la nécessité d'avoir un tri en temps réel pour une utilisation dans le contexte BMI [Lew98]. Au final, les neurones avec une forme d'impulsion clairement identifiable sont isolés et regroupés.

L'activité neurale intra-corticale est habituellement enregistrée avec des fréquences d'échantillonnage comprises entre 30 kHz et 50 kHz. Comme le nombre de canaux peut lui aussi être très élevé (128 électrodes ou plus), cela génère une importante quantité de données à traiter en temps réel. Pour un système embarqué ou implanté, il faut pour transmettre ces données en temps réel une bande-passante sans-fil très importante de l'ordre de plusieurs Mbps. Par conséquent, les données doivent être compressées avant la transmission, mais avec peu d'énergie disponible : proposer une méthode de compression à bas coût est alors pertinent. Le problème de réduction de données est en partie traité par la détection d'impulsions : il existe des algorithmes qui permettent la détection d'impulsion par seuil adaptatif en temps réel [BBL+09]. On propose d'utiliser la projection sur une matrice binaire aléatoire comme méthode de compression, avec comme objectif final de permettre la classification des données. En effet, la classification des impulsions est généralement effectuée après l'acquisition de données par un système BMI, afin de déterminer quel neurone a déclenché quelle impulsion.

L'expérience décrite dans la suite de ce chapitre permet alors de vérifier les possibilités de classification après projection sur une matrice aléatoire binaire.

7.1.2 Description des données

Les données utilisées sont des données neurales simulées, rendues disponibles par les auteurs de [QQNBS04]. La méthode de génération des données est résumée ici : le bruit ambiant est simulé en utilisant des formes d'impulsion issues d'une base d'environ 600 formes enregistrées et moyennées, placées aléatoirement en temps et en amplitude. Un train de données de 3 formes d'impulsion est ensuite ajouté avec une amplitude normalisée. Les données sont simulées à 96kHz et interpolées de façon à ce que les impulsions soient placées de manière continue dans le temps (à la précision de la machine près). Ensuite, les données sont sous-échantillonnées à 24 kHz pour imiter les conditions réelles d'enregistrement. La description détaillée de la méthode de génération est disponible dans l'article original [QQNBS04].

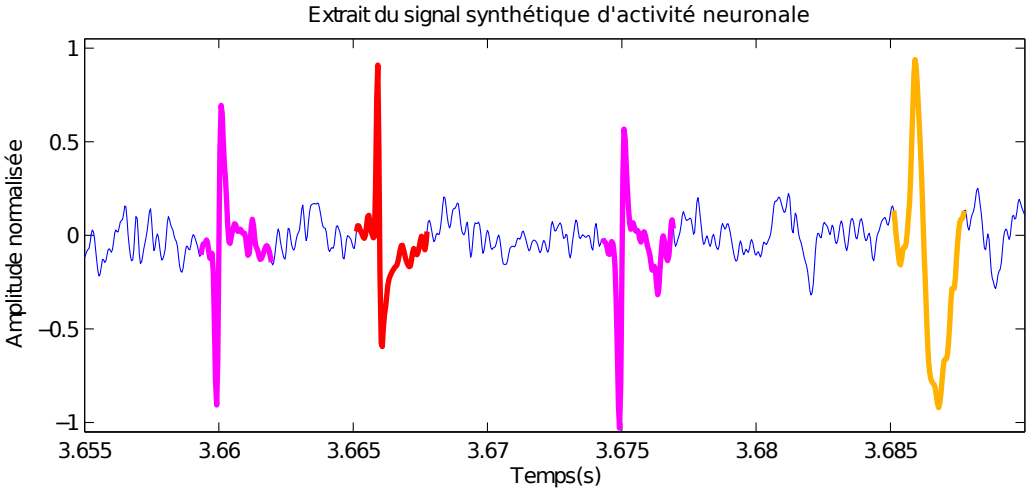

FIGURE 7.1 – Extrait du signal issu des données simulées, échantillonné à 24 kHz, où les impulsions sont surlignées de couleurs différentes selon leur cluster d'appartenance.

On utilisera par la suite un signal simulé d'une longueur de 10 secondes, contenant 507 impulsions. L'emplacement et le cluster d'appartenance de chaque impulsion sont connus, et on dispose aussi de chaque impulsion extraite du signal synchronisée sur son point d'amplitude maximale (fig. 7.2, 32 échantillons par impulsion). La figure 7.1 montre un extrait du signal, où les impulsions sont surlignées, d'une couleur correspondant à leur cluster d'appartenance. Une Analyse en Composantes Principales (ACP) effectuée sur les impulsions montre 3 groupes bien distincts (fig. 7.3) sur les deux premières composantes de l'ACP, permettant une classification aisée. Cela signifie qu'idéalement, il suffirait de deux projections pour effectuer la classification finale. Cependant, il n'est pas possible de connaître a priori les deux axes principaux, ou alors cela nécessiterait le stockage de toutes les données et le calcul de l'ACP, ce qui est exclu dans ce contexte de système embarqué léger.

7.1.3 Méthodes

7.1.3.1 Compression

La compression est effectuée à la manière du codage parcimonieux, en utilisant comme matrice de codage une matrice aléatoire binaire ± 1 équiprobable. Dans une situation réelle, le signal est échantillonné et numérisé sous forme d'entier signé. L'opération de projection est alors facile (cf chapitre 5) : il s'agit d'un changement de signe binaire (changement du bit de signe) et d'additions binaires.

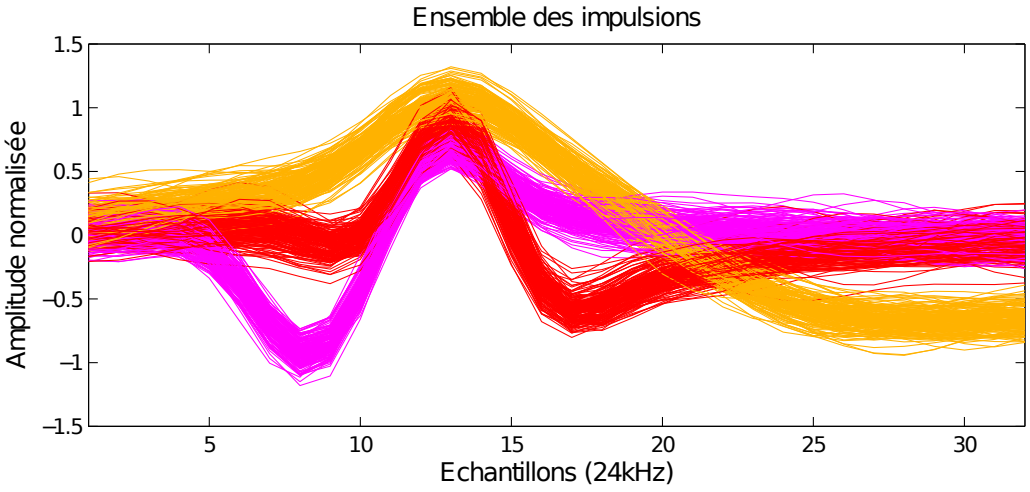

FIGURE 7.2 – Superposition des 507 impulsions synchronisées, chaque groupe ayant sa propre couleur.

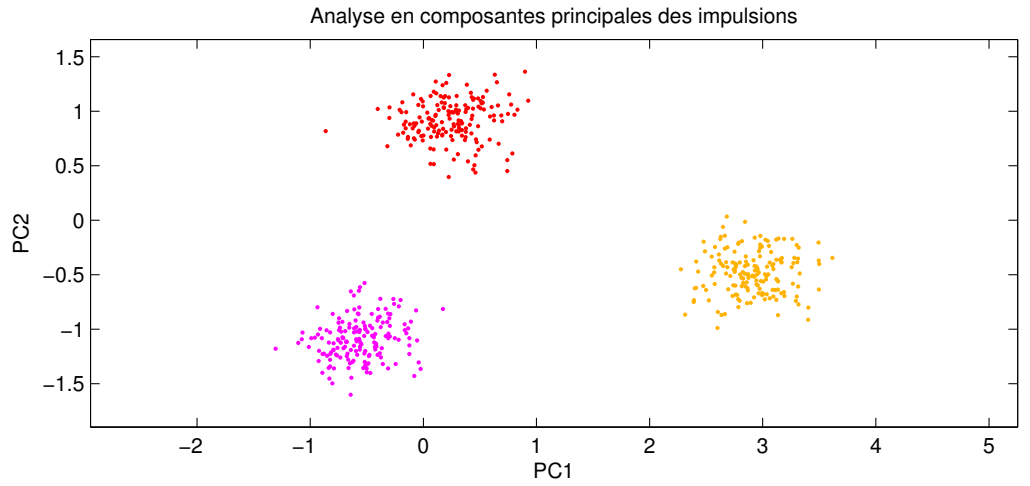

FIGURE 7.3 – Projection sur les deux premières composantes principales des impulsions, les couleurs correspondant au cluster (connu) de l'impulsion correspondante.

7.1.3.2 Classification

La classification est réalisée directement sur les données compressées à l'aide de l'algorithme k-means [Llo82]. C'est un algorithme qui vise à réaliser un partitionnement des données $\mathbf{y}$ en k ensembles $\mathbf{S} = \{S_1, S_2, ..., S_k\}$ tels que :

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{j \in S_i} \|\mathbf{y}_j - \mu_i\|_2^2,$$

où les μ_i sont les moyennes des ensembles S_i. Habituellement l'algorithme fonctionne de manière itérative, en associant chaque vecteur à la moyenne la plus proche, puis en mettant

à jour les moyennes en fonction des vecteurs sélectionnés pour chaque groupe. Il est donc nécessaire d'initialiser ces moyennes. Pour effectuer ceci de manière automatique, on a utilisé une méthode inspirée par [GMCH11]. L'idée est de construire un Arbre Couvrant de Poids Minimal (ACDM ; en anglais MST, *Minimum Spanning Tree*), au sens de la distance euclidienne, à partir d'un point choisi au hasard. Chaque nouveau point ajouté à l'arbre est celui pour lequel il faut la plus courte distance pour le joindre à un des points appartenant déjà à l'arbre. En conservant la séquence des distances de chaque ajout, il est possible de distinguer des groupes en effectuant un seuillage. Les moyennes de ces groupes servent alors d'initialisation pour l'algorithme k-means.

7.1.4 Résultats

Les simulations ont été répétées pour différentes valeurs du nombre m de lignes de la matrice de projection Φ, en utilisant à chaque fois 1000 matrices binaires tirées aléatoirement.

7.1.4.1 Initialisation du partitionnement

Pour initialiser les moyennes, le niveau de seuillage des distances d'ajout dans l'ACDM est déterminé a posteriori et fixé à la valeur moyenne des distances plus un écart-type, et la taille minimale pour former un groupe est fixée à $\#S/6$, où $\#S$ est le nombre total d'impulsions, ici 507. Cela signifie qu'il y a au plus 6 groupes, ce qui est cohérent avec les enregistrements physiologiques où un nombre maximal de 4 groupes est généralement observé. La figure 7.4 montre un exemple de séquence de distances pour la construction d'un arbre, ainsi que le seuil fixé pour constituer les groupes. Les parties de l'arbre correspondant aux groupes retenus, sont montrées sur la figure 7.5.

Nombre de projections m	4	5	6	7	8	9	10	11	12
Taux de compression	8	6.4	5.33	4.57	4	3.56	3.2	2.91	2.67
Proportion d'impulsions mal classifiées	5.09%	2.38%	0.81%	0.48%	0.25%	0.07%	0.03%	0.02%	0.02%
Moins de 0,5% d'erreur	69.4%	83.1%	91.8%	95.2%	97.6%	98%	99%	99.6%	99.6%
Nombre moyen de groupes	2.855	2.934	2.979	2.988	2.994	2.999	3	3	3
Proportion de réalisations avec moins de 3 groupes	14.3%	6.6%	2.1%	1.2%	0.6%	0.1%	0%	0%	0%

TABLE 7.1 – Performances de la classification en fonction du nombre de projections m, $N = 32$.

La figure 7.6 montre le pourcentage moyen d'erreurs (obtenu sur 1000 réalisations différentes de la matrice, pour chaque valeur de m) en fonction du nombre de projections

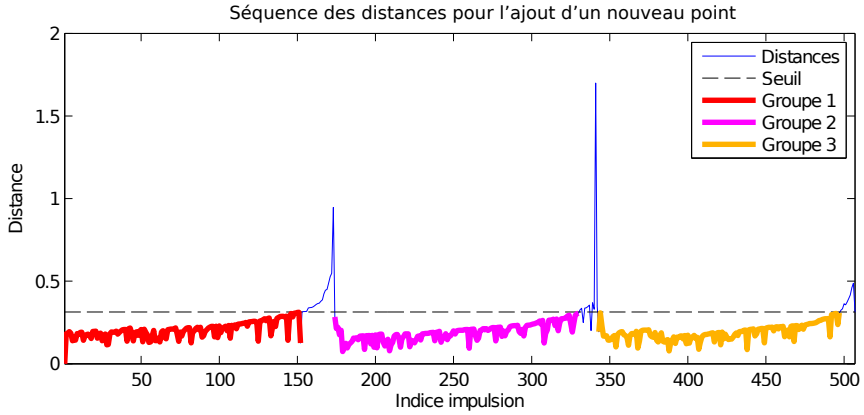

FIGURE 7.4 – Séquence des distances pour ajouter un nouveau point à l'arbre couvrant de poids minimal. Le niveau de seuillage est indiqué par la ligne horizontale et les groupes identifiés sont surlignés en couleurs différentes.

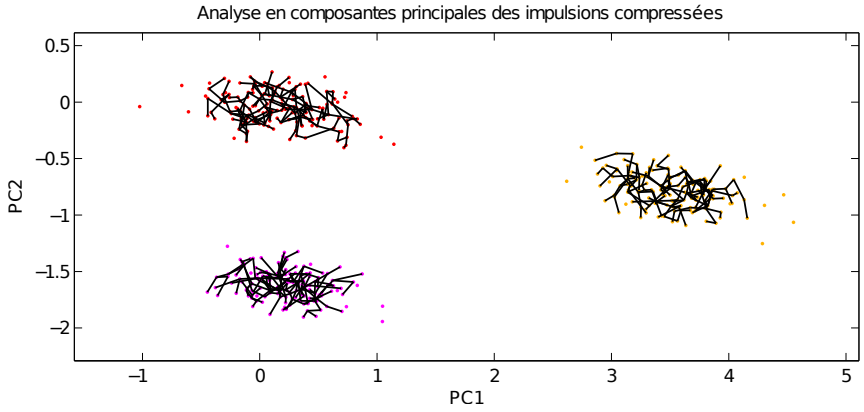

FIGURE 7.5 – Arbres couvrants de distance minimale pour les groupes identifiés.

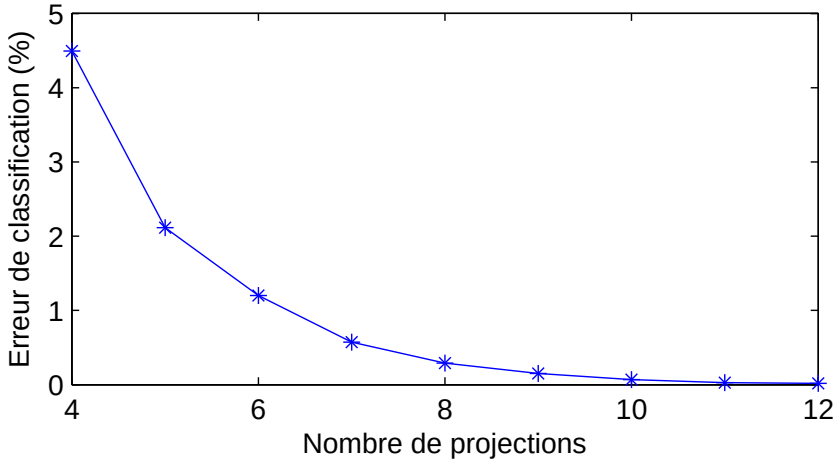

FIGURE 7.6 – Nombre moyen d'impulsions mal classées en fonction du nombre de projections m, pour des signaux de $N = 32$ points.

m variant de 4 à 12, soit des taux de compression variant de 8 à $2,67$ respectivement. On constate que pour $m \geq 6$, l'erreur est telle qu'il y a moins de 5 impulsions dont le classement est erroné. Pour $m \geq 9$, il y a en moyenne moins d'une erreur de classification.

Un exemple de bon résultat pour un fort taux de compression (8, i.e. $m = 4$) est visible sur la figure 7.7 : on y voit la projection sur deux composantes principales [1], et trois ensembles qui se distinguent, toutefois moins bien séparés (ou plus répandus) que sur la figure 7.3. Cet exemple correspond au résultat obtenu dans la majorité des cas comme le montre la seconde ligne du tableau 7.1, même dans le cas le plus défavorable ($m = 4$).

Cependant, pour les forts taux de compression, il arrive (environ 30% des cas) qu'il y ait un mélange entre deux ensembles, comme sur l'exemple de la figure 7.8. Cela représente la majorité des cas où des erreurs sont relevées (85% quand $m = 4$, 100% quand $m > 10$), les autres cas présentant des erreurs dans les trois groupes.

La pire des situations qui puisse se présenter est lorsque deux groupes sont complètement mélangés, comme sur la figure 7.9, ce qui conduit à un taux d'erreur très important puisque chaque ensemble représente environ un tiers de la population. Cela se produit dans $14,1$% des cas lorsque $m = 4$ (il arrive même que les trois groupes soient mélangés, uniquement pour 0,2% des situations). Cela s'améliore fortement quand m augmente, et pour $m \geq 10$, cette situation ne se présente plus. Dans les cas où ce mélange de groupe se produit, les deux groupes mélangés sont en général (plus de 83% des cas) très bien séparés du troisième. Les diagrammes en barre des figure 7.7 – 7.9 montrent que la classification parfaite est très fréquente, si ce n'est dans le cas le plus défavorable ($m = 4$). Dans la majorité des cas, les résultats sont au moins aussi bons que ceux visibles sur la figure 7.8.

Le tableau 7.1 résume les résultats de classification obtenus par la simulation. La première ligne, qui reprend les résultats de la figure 7.6, indique la proportion de signaux mal classifiés, en moyenne sur les 1000 réalisations, alors que la seconde ligne indique la proportion de réalisations où il y avait au plus, deux erreurs de classification (soit une erreur inférieure à 0,5%). Quand $m \geq 6$, cela se produit dans plus de 90% des cas, et il y a moins de 4 signaux dont le groupe attribué est erroné. Les deux dernières lignes traitent du nombre de groupes détectés : on remarque que si c'est un problème occasionnel pour les forts taux de compressions, lorsque $m \geq 10$ il n'y a plus d'erreur à ce niveau là, et les trois groupes sont toujours bien discernés.

7.1.5 Conclusion

Les résultats de cette expérience permettent de constater qu'il est possible de travailler sur les signaux compressés en ayant des performances acceptables. On ne multiplie que d'un facteur 3 le nombre de projections permettant d'effectuer une classification par rapport à ce

1. L'ACP et cette projection ne sont effectuées que par souci de lisibilité. L'opération de classification est réalisée sur les m composantes des signaux projetés.

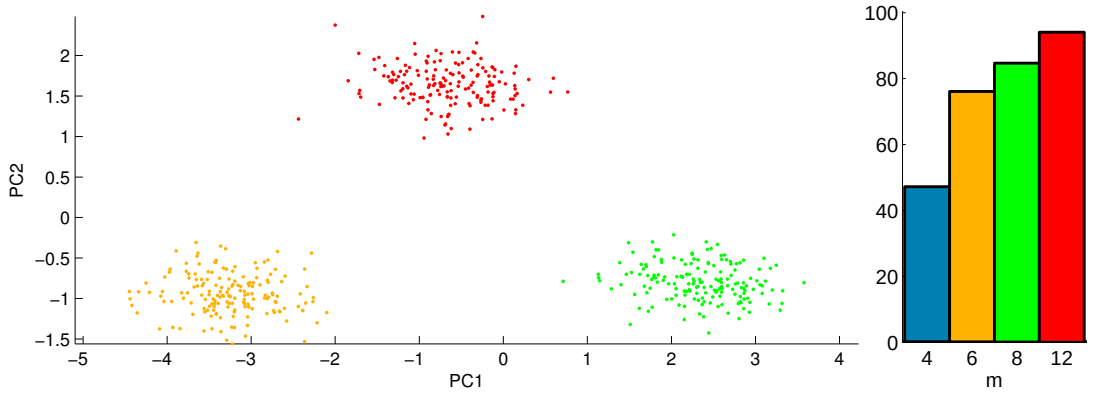

FIGURE 7.7 – À gauche : projection sur les deux premières composantes principales des signaux compressés ($m = 4$, $N = 32$), avec une classification parfaite. À droite : la proportion des réalisations qui font aussi bien, pour $m = 4$, 6, 8 et 12.

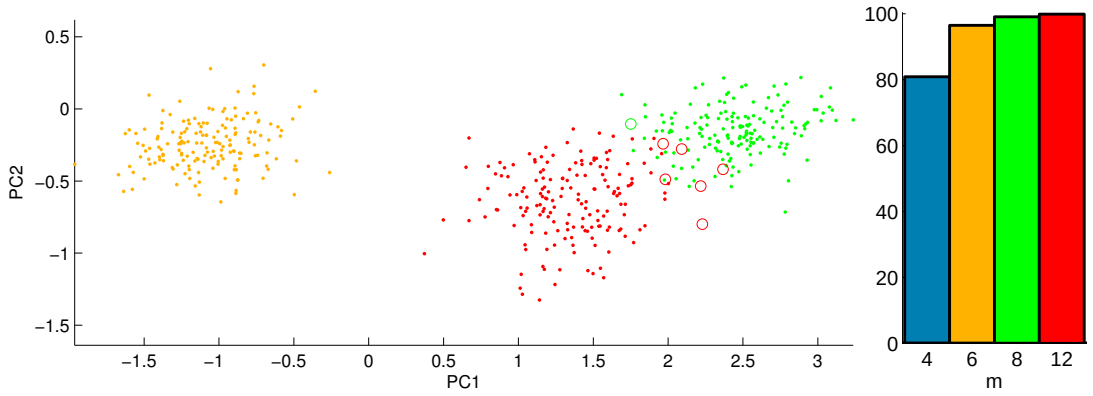

FIGURE 7.8 – À gauche : projection sur les deux premières composantes principales des signaux compressés ($m = 4$, $N = 32$), avec une classification présentant 7 erreurs. À droite : la proportion des réalisations qui font aussi bien ou mieux, pour $m = 4$, 6, 8 et 12.

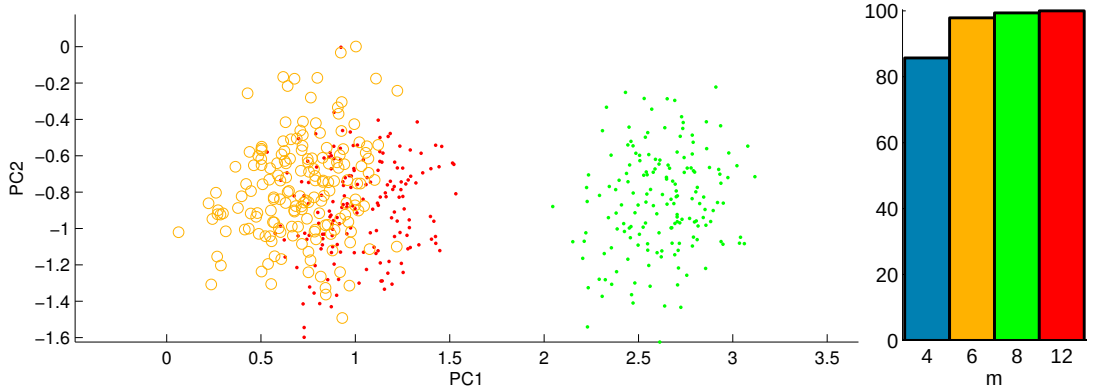

FIGURE 7.9 – À gauche : projection sur les deux premières composantes principales des signaux compressés ($m = 4$, $N = 32$), lorsque deux classes sont confondues. À droite : la proportion des réalisations qui font mieux, pour $m = 4$, 6, 8 et 12.

qui serait nécessaire a posteriori à l'aide d'une ACP, mais en se passant de ces lourds calculs et de grandeurs réelles. On remarque que si on se réfère aux résultats expérimentaux de reconstruction du chapitre 4, il serait certainement impossible de reconstruire les signaux originaux avec seulement 4 ou 6 projections, en ayant un dictionnaire de représentation permettant de décomposer le signal sur 2 ou 3 composantes. Toutefois, il ne faut pas oublier que ces simulations ont leurs limites : la situation était propice au bon résultat, puisque le rapport signal à bruit est très bon et qu'il est facile de distinguer les classes avant la projection. Comme la séparation des clusters est dégradée par la projection, on peut penser que dans une situation moins avantageuse, les résultats seraient moins bons.

7.2 Étude bibliographique

La question des possibilités de classification après une projection sur des vecteurs aléatoires (qu'on dénommera simplement projection aléatoire) a déjà été soulevée dans la littérature, notamment dans une approche de réduction de dimensionnalité. Les travaux de Dasgupta [Das99, Das00] s'intéressent à l'évolution de mixtures de gaussiennes et leur évolution lorsque celles-ci sont projetées sur un espace aléatoire de dimension inférieure. On retrouve alors la situation $y = \Phi x$ typique du codage parcimonieux, où les éléments de la matrice Φ sont issus d'un processus aléatoire gaussien, indépendants entre eux. Cependant, il n'y a plus d'hypothèse de décomposition parcimonieuse de x Ces travaux indiquent notamment la manière dont évolue la séparation entre deux groupes.

D'autres travaux de recherche s'intéressent aussi aux possibilités de classification et de clustering à partir de projections aléatoires. On peut citer les travaux de [BV06], [FB03] où l'idée est de réaliser plusieurs projections aléatoires suivies à chaque fois d'une classification (souvent à l'aide de l'algorithme k-means) et d'agglomérer les résultats. On s'éloigne alors des problématiques du codage parcimonieux puisque la répétition des projections n'entraine plus de compression finale. Ces travaux se placent essentiellement dans l'optique des problèmes de grandes dimensions, où il est nécessaire de réduire la taille des données pour pouvoir les traiter (ce qui n'est pas le cas de notre expérience). On remarque aussi que ces travaux s'appuient sur les résultats de Johnson et Lindenstrauss [JL84], ce qui rappelle la démonstration de la Restricted Isometry Property de [BDDW08] et établit donc un lien avec le codage parcimonieux. Ce lien est aussi repris par [BW09] où apparaît une propriété proche de la RIP et le lemme de Johnson-Lindestrauss sous la forme suivante :

Lemme 1 *Soit Q un ensemble fini de $\#Q$ points de $\mathbb{R}^N$. Soient $0 < \varepsilon < 1$ et $\beta > 0$. Soit Φ un projecteur orthogonal aléatoire de $\mathbb{R}^N$ dans $\mathbb{R}^m$ (c'est-à-dire une matrice de taille $m \times N$ dont les lignes sont orthogonales entre elles) avec*

$$m \geq \left(\frac{4 + 2\beta}{\varepsilon^2/2 + \varepsilon^3/3} \right) \ln \#Q. \tag{7.1}$$

Si $m \leq N$, alors pour tout $x, y \in Q$, la propriété suivante est vraie avec une probabilité supérieure à $1 - (\#Q)^{-\beta}$:

$$(1-\varepsilon)\sqrt{\frac{m}{N}} \leq \frac{\|\Phi x - \Phi y\|_2}{\|x-y\|_2} \leq (1+\varepsilon)\sqrt{\frac{m}{N}}. \tag{7.2}$$

Le choix fait par les auteurs de [BW09] d'avoir un projecteur orthogonal conduit à la présence du facteur $\frac{m}{N}$, mais une simple renormalisation par $\frac{N}{m}$ conduit à une formule beaucoup plus proche de l'expression de la RIP [Can08] :

$$(1-\varepsilon) \leq \frac{\|\Phi x - \Phi y\|_2}{\|x-y\|_2} \leq (1+\varepsilon). \tag{7.3}$$

On retrouve aussi souvent dans la littérature la forme

$$(1-\delta) \leq \frac{\|\Phi x\|_2^2}{\|x\|_2^2} \leq (1+\delta) \tag{7.4}$$

pour les matrices aléatoires à colonnes normalisées. Les deux équations (7.4) et (7.3) ne sont pas équivalentes, mais il existe un lien simple entre les deux : si $\delta < 1$ alors

$$(7.4) \Leftrightarrow \sqrt{(1-\delta)} \leq \frac{\|\Phi x\|_2}{\|x\|_2} \leq \sqrt{(1+\delta)} \Rightarrow (1-\delta) \leq \frac{\|\Phi x\|_2}{\|x\|_2} \leq (1+\delta) \Leftrightarrow (7.3); \tag{7.5}$$

et de manière converse, si $\varepsilon < 1$, on a :

$$\begin{aligned}(7.3) \Leftrightarrow (1-2\varepsilon+\varepsilon^2) \leq \frac{\|\Phi x\|_2^2}{\|x\|_2^2} \leq (1+2\varepsilon+\varepsilon^2) \\ \Rightarrow (1-\mathbf{3}\varepsilon) \leq \frac{\|\Phi x\|_2^2}{\|x\|_2^2} \leq (1+\mathbf{3}\varepsilon),\end{aligned} \tag{7.6}$$

ce qui permet d'avoir un lien entre les deux équations.

D'après l'énoncé du lemme 1, on peut établir une valeur minimale pour le nombre de projections nécessaires pour vérifier l'équation 7.2, sous la forme

$$m_{min} = \left(\frac{4+2\beta}{\varepsilon^2/2 + \varepsilon^3/3}\right) \ln \#Q, \tag{7.7}$$

où $\#Q$ représente le nombre de vecteurs x considérés, et β est choisi en fonction de la probabilité minimale voulue P_{min}, par l'expression $\beta = -\frac{\ln 1-P}{\ln \#Q}$ La figure 7.10 montre que ce nombre minimal est assez élevé, même si on autorise un ε grand, c'est-à-dire une large déviation par rapport à l'isométrie. Comme par hypothèse $N > m$, cela signifie qu'on ne peut pas s'appuyer sur ce lemme dans les cas où les dimensions sont petites.

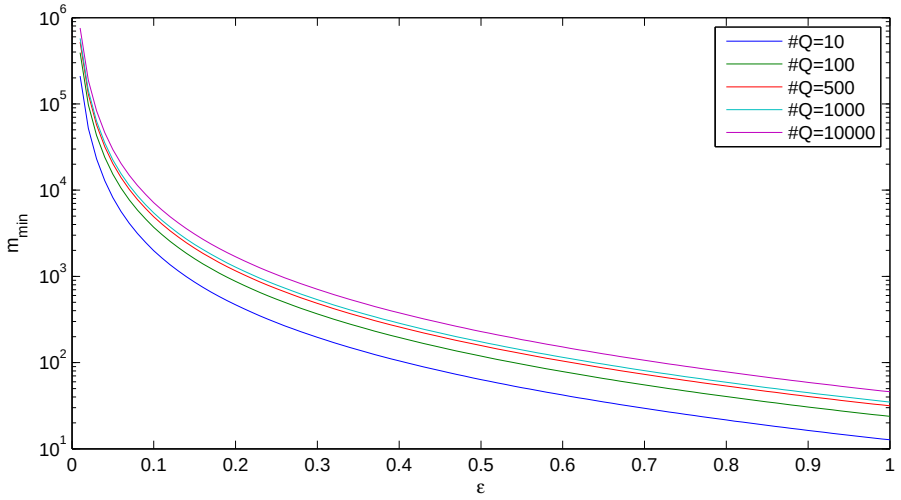

FIGURE 7.10 – Variation du nombre minimal de projections nécessaires $m_{min} = \left(\frac{4+2\beta}{\varepsilon^2/2+\varepsilon^3/3}\right) \ln \#Q$ en fonction de ε, avec β tel que $P > 0.5$ et pour différentes valeurs de $\#Q$.

7.3 Conservation de la norme par projection sur une matrice aléatoire

7.3.1 Résultat proposé

D'autres auteurs ont remarqué les difficultés de calculer la constante d'isométrie restreinte d'une matrice et ont proposé des méthodes (StRIP, ExRIP) qui permettent d'évaluer la probabilité qu'une matrice d'un type donné vérifie la RIP. En s'inspirant notamment des résultats de [ME09b], nous proposons le résultat suivant :

Proposition 2 *Soit Φ une matrice $m \times N$ dont les éléments $\phi_{i,j}$ prennent les valeurs $\pm\frac{1}{\sqrt{m}}$, tels que $P(\phi_{i,j} = \frac{1}{\sqrt{m}}) = P(\phi_{i,j} = -\frac{1}{\sqrt{m}}) = 0.5$. Soient $x \in \mathbb{R}^N$ issu d'un processus stochastique et $\delta > 0$, alors :*

$$P\left\{\left|\frac{\|\Phi x\|^2}{\|x\|^2} - 1\right| \leq \delta\right\} \geq 1 - \frac{2}{m\delta^2}\left(1 - \mathbb{E}\left\{\frac{\sum_{a=1}^N x_a^4}{\|x\|^4}\right\}\right). \tag{7.8}$$

Ce résultat s'appuie sur l'inégalité de Bienaymé-Chebychev et découle des propriétés de la matrice, principalement du fait que l'espérance de la valeur de chaque élément est nulle, et de l'indépendance entre les éléments $\phi_{i,j}$.

Le terme $\mathbb{E}\left\{\frac{\sum_{a=1}^N x_a^4}{\|x\|^4}\right\}$ est toujours positif et inférieur à 1 (si on exclut le cas trivial où x ne comporte qu'un seul élément non nul). En effet, $\frac{\sum_{a=1}^N x_a^4}{\|x\|^4} = \frac{\sum_{a=1}^N x_a^4}{\sum_{a=1}^N x_a^4 + \sum_{a=1}^N \sum_{b=1,b\neq a}^N x_a^2 x_b^2}$ $= \frac{1}{1+\alpha}$ avec $\alpha > 0$. Dans le cas des signaux utilisés dans la section précédente (fig. 7.2), la valeur moyenne calculée vaut en effet ≈ 0.1067. Le tableau 7.2 donne quelques valeurs

de cette espérance [2] pour les distributions uniforme et normale, où l'on constate aussi que $\mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|^4}\right\} \approx \frac{\alpha}{N}$, avec $\alpha = 1,8$ pour la distribution uniforme, et $\alpha = 3$ pour la distribution normale.

	N=4	N=8	N=16	N=32	N=64	N=128	N=256	N=512	N=1024
Uniforme	0.4285	0.2252	0.113	0.0564	0.0282	0.0141	0.007	0.0035	0.0018
Normale	0.4997	0.3003	0.1666	0.0883	0.0455	0.0231	0.0116	0.0058	0.0029

TABLE 7.2 – Valeur de $\mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|^4}\right\}$ selon la dimension N du signal, pour des lois de distribution uniforme (sur $[-0,5\,;\,0,5]$) et normale ($\mathcal{N} \sim (0,1)$).

On peut donc définir une borne moins stricte, en considérant ce terme comme nul :

$$P\left\{\left|\frac{\|\Phi x\|^2}{\|x\|^2} - 1\right| \leq \delta\right\} \geq 1 - \frac{2}{m\delta^2}. \tag{7.9}$$

En injectant (7.5) dans (7.8), on obtient alors

$$P\left\{(1-\varepsilon) \leq \frac{\|\Phi x\|_2}{\|x\|_2} \leq (1+\varepsilon)\right\} \geq 1 - \frac{2}{m\varepsilon^2}\left(1 - \mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|_2^4}\right\}\right) \geq 1 - \frac{2}{m\varepsilon^2}. \tag{7.10}$$

7.3.2 Démonstration

Le résultat se démontre en calculant $\mathbb{E}\{Z^2\}$ et $\mathbb{E}\{Z^4\}$, où $Z = \frac{\|\Phi x\|}{\|x\|}$, puis en appliquant l'inégalité de Bienaymé-Chebyshev :

$$P\{|Z^2 - \mathbb{E}\{Z^2\}| \leq \delta\} \geq 1 - \frac{\mathbb{E}\{Z^4\} - \mathbb{E}\{Z^2\}^2}{\delta^2}.$$

2. Valeurs obtenues par simulation sur 100000 signaux aléatoires pour chaque valeur de N.

7.3.2.1 Terme de deuxième ordre

$$\text{Calculons } \mathbb{E}\left\{Z^2\right\} = \mathbb{E}\left\{\frac{\|\Phi x\|^2}{\|x\|^2}\right\} = \mathbb{E}\left\{\frac{\sum_{i=1}^{m}\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)^2}{\|x\|^2}\right\} = \sum_{i=1}^{m}\mathbb{E}\left\{\frac{\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)^2}{\|x\|^2}\right\}$$

$$= \sum_{i=1}^{m}\mathbb{E}\left\{\frac{\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)\left(\sum_{k=1}^{N}\phi_{i,k}x_k\right)}{\|x\|^2}\right\} = \sum_{i=1}^{m}\mathbb{E}\left\{\frac{\left(\sum_{j=1}^{N}\phi_{i,j}^2x_j^2\right) + \left(\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq j}}^{N}\phi_{i,j}\phi_{i,k}x_jx_k\right)}{\|x\|^2}\right\}$$

$$= \sum_{i=1}^{m}\left(\sum_{j=1}^{N}\mathbb{E}\left\{\phi_{i,j}^2\right\}\mathbb{E}\left\{\frac{x_j^2}{\|x\|^2}\right\} + \sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq j}}^{N}\mathbb{E}\left\{\phi_{i,j}\right\}\mathbb{E}\left\{\phi_{i,k}\right\}\mathbb{E}\left\{\frac{x_jx_k}{\|x\|^2}\right\}\right)$$

Mais puisque $\mathbb{E}\left\{\phi_{i,j}^2\right\} = \frac{1}{m}$ et $\mathbb{E}\left\{\phi_{i,j}\right\} = 0$, on obtient alors :

$$\mathbb{E}\{Z^2\} = \mathbb{E}\left\{\frac{\sum_{i=1}^{m}\left(\sum_{j=1}^{N}\frac{1}{m}x_j^2\right) + 0}{\|x\|^2}\right\} = \mathbb{E}\left\{\frac{\sum_{i=1}^{m}\left(\frac{1}{m}\|x\|_2\right)}{\|x\|^2}\right\} = 1$$

7.3.2.2 Terme d'ordre 4

$$\text{Calculons maintenant } \mathbb{E}\{Z^4\} = \mathbb{E}\left\{\frac{\|\Phi x\|^4}{\|x\|^4}\right\} = \mathbb{E}\left\{\frac{\left(\sum_{i=1}^{m}\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)^2\right)^2}{\|x\|^4}\right\}$$

$$= \mathbb{E}\left\{\frac{\left(\sum_{i=1}^{m}\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)^2\right)\left(\sum_{a=1}^{m}\left(\sum_{b=1}^{N}\phi_{a,b}x_b\right)^2\right)}{\|x\|^4}\right\} = \mathbb{E}\left\{\frac{\sum_{i=1}^{m}\sum_{a=1}^{m}\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)^2\left(\sum_{b=1}^{N}\phi_{a,b}x_b\right)^2}{\|x\|^4}\right\}$$

$$= \mathbb{E}\left\{\frac{\sum_{i=1}^{m}\sum_{a=1}^{m}\left(\sum_{j=1}^{N}\sum_{l=1}^{N}\phi_{i,j}\phi_{i,l}x_jx_l\right)\left(\sum_{b=1}^{N}\sum_{c=1}^{N}\phi_{a,b}\phi_{a,c}x_bx_c\right)}{\|x\|^4}\right\}$$

$$= \sum_{i=1}^{m}\sum_{a=1}^{m}\sum_{j,l,b,c=1}^{N} \mathbb{E}\left\{\phi_{i,j}\phi_{i,l}\phi_{a,b}\phi_{a,c}\right\} \mathbb{E}\left\{\frac{x_j x_l x_b x_c}{\|x\|^4}\right\}$$

On renomme les indices, pour une meilleure lisibilité :

$$\mathbb{E}\{Z^4\} = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{a,b,c,d=1}^{N} \mathbb{E}\left\{\phi_{i,a}\phi_{i,b}\phi_{j,c}\phi_{j,d}\right\} \mathbb{E}\left\{\frac{x_a x_b x_c x_d}{\|x\|^4}\right\}$$

Distinguons maintenant les différents cas possibles pour a, b, c, d ainsi que pour i et j dans le terme $\mathbb{E}\left\{\phi_{i,a}\phi_{i,b}\phi_{j,c}\phi_{j,d}\right\}$

- $a = b = c = d$:
  - $i \neq j$:

$$A = \sum_{\substack{a,b,c,d=1\\ a=b=c=d}}^{N} \mathbb{E}\left\{\phi_{i,a}\phi_{i,b}\phi_{j,c}\phi_{j,c}\right\} \mathbb{E}\left\{\frac{x_a x_b x_c x_d}{\|x\|^4}\right\} = \sum_{a=1}^{N} \mathbb{E}\left\{\phi_{i,a}^2\phi_{j,a}^2\right\} \mathbb{E}\left\{\frac{x_a^4}{\|x\|^4}\right\}$$

$$= \sum_{a=1}^{N} \mathbb{E}\left\{\phi_{i,a}^2\right\} \mathbb{E}\left\{\phi_{j,a}^2\right\} \mathbb{E}\left\{\frac{x_a^4}{\|x\|^4}\right\} = \frac{1}{m^2}\mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|^4}\right\}.$$

  - $i = j$

$$B = \sum_{\substack{a,b,c,d=1\\ a=b=c=d}}^{N} \mathbb{E}\left\{\phi_{i,a}\phi_{i,b}\phi_{i,c}\phi_{i,c}\right\} \mathbb{E}\left\{\frac{x_a x_b x_c x_d}{\|x\|^4}\right\}$$

$$= \sum_{a=1}^{N} \mathbb{E}\left\{\phi_{i,a}^4\right\} \mathbb{E}\left\{\frac{x_a^4}{\|x\|^4}\right\} = \frac{1}{m^2}\mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|^4}\right\} = A.$$

- S'il y en a un différent des 3 autres, alors le terme $\phi_{.,.}$ est indépendant des autres, et l'espérance est alors nulle.
- Il reste alors les cas où a, b, c et d forment deux paires, ce qui donne trois combinaisons à traiter :
  - $a = b$ et $c = d$, $a \neq c$, le résultat ne dépend pas des indices i et j :

$$C = \sum_{\substack{a,b,c,d=1\\ a=b\neq c=d}}^{N} \mathbb{E}\left\{\phi_{i,a}\phi_{i,b}\phi_{j,c}\phi_{j,d}\right\} \mathbb{E}\left\{\frac{x_a x_b x_c x_d}{\|x\|^4}\right\}$$

$$= \sum_{\substack{a,b=1\\ a\neq b}}^{N} \mathbb{E}\left\{\phi_{i,a}^2\phi_{j,b}^2\right\} \mathbb{E}\left\{\frac{x_a^2 x_b^2}{\|x\|^4}\right\} = \frac{1}{m^2}\mathbb{E}\left\{\frac{\sum_{\substack{a,b=1\\ a\neq b}}^{N} x_a^2 x_b^2}{\|x\|^4}\right\}$$

  - $a = c$, $b = d$ (équivalent au cas $a = d$, $b = c$) et $i = j$ (si $i \neq j$, l'espérance est

nulle) :

$$D = \sum_{\substack{a,b,c,d=1 \\ a=c\neq b=d}}^{N} \mathbb{E}\left\{\phi_{i,a}\phi_{i,b}\phi_{j,c}\phi_{j,d}\right\} \mathbb{E}\left\{\frac{x_a x_b x_c x_d}{\|x\|^4}\right\}$$

$$= \sum_{\substack{a,b=1 \\ a\neq b}}^{N} \mathbb{E}\left\{\phi_{i,a}^2\phi_{i,b}^2\right\} \mathbb{E}\left\{\frac{x_a^2 x_b^2}{\|x\|^4}\right\} = \frac{1}{m^2}\mathbb{E}\left\{\frac{\sum_{\substack{a,b=1 \\ a\neq b}}^{N} x_a^2 x_b^2}{\|x\|^4}\right\}$$

En comptant le nombre d'occurrences pour chaque cas A, B, C et D et sachant que $A = B$ et $C = D$, on a alors $\mathbb{E}\{Z^4\} = m(m-1)A+mB+m^2C+2mD = m^2A+(m^2+2m)C$, c'est-à-dire :

$$\mathbb{E}\{Z^4\} = \mathbb{E}\left\{\frac{1}{\|x\|^4}\sum_{a=1}^{N} x_a^4 + (1+\frac{2}{m})\frac{1}{\|x\|^4}\sum_{\substack{a,b=1 \\ a\neq b}}^{N} x_a^2 x_b^2\right\}.$$

Mais puisque $\sum_{\substack{a,b=1 \\ a\neq b}}^{N} x_a^2 x_b^2 = \|x\|^4 - \sum_{a=1}^{N} x_a^4$, cela donne $\mathbb{E}\{Z^4\} = 1+\frac{2}{m}\left(1 - \mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|^4}\right\}\right)$.

On peut donc appliquer l'inégalité de Bienaymé-Chebyshev :

$$P\{|Z^2 - \mathbb{E}\{Z^2\}| \leq \delta_k\} \geq 1 - \frac{Var\{Z^2\}}{\delta_k^2} = 1 - \frac{\mathbb{E}\{Z^4\} - \mathbb{E}\{Z^2\}^2}{\delta_k^2};$$

qui conduit au résultat final :

$$P\left\{\left|\frac{\|\Phi x\|^2}{\|x\|^2} - 1\right| \leq \delta_k\right\} \geq 1 - \frac{2}{m\delta_k^2}\left(1 - \mathbb{E}\left\{\frac{\sum_{a=1}^{N} x_a^4}{\|x\|^4}\right\}\right).$$

7.3.3 Comparaison avec le lemme 1

À partir de l'équation (7.10), il est facile de déterminer un nombre minimal de projections nécessaires pour atteindre une probabilité minimale donnée :

$$m_{min} = \frac{2}{\varepsilon^2(1-P)}. \tag{7.11}$$

La figure 7.11 montre la comparaison entre cette valeur, et celle calculée selon l'énoncé du lemme 1 (éq. 7.7). On constate que pour une probabilité basse, le résultat proposé permet d'obtenir la même probabilité minimale avec un nombre de projections bien inférieur à celui requis selon les conditions du lemme 1. En revanche, pour des probabilités plus proches de 1, la différence s'estompe, puis la condition du lemme 1 sur m devient moins forte que celle de la proposition 2, lorsque la probabilité minimale recherchée est asymptotiquement proche de 1.

La figure 7.12 montre l'avantage de la proposition 2 dans le cas de petites matrices

(ici, $N = 32$) : La borne calculée par l'équation 7.10 n'est pas très bonne[3], puisque la valeur calculée n'est supérieure à 0 que pour de fortes distorsions par rapport à l'isométrie. Cependant, le nombre de projections requises par le lemme 1 pour atteindre ces probabilités est bien supérieur à N, ce qui contredit l'hypothèse $m < N$ du lemme et du principe étudié ici, à savoir la compression de données.

Puisque notre résultat s'appuie sur l'inégalité de Bienaymé-Chebyshev, qui est réputée pour ne pas être très serrée, nous avons approché le problème par l'inégalité de Chernoff, inspirés par [PL11]. Cependant, les approximations devant être faites pour résoudre le calcul font que la borne n'est pas fiable dès que le nombre de projections est supérieur à 4. Ces résultats sont présentés en annexe E.

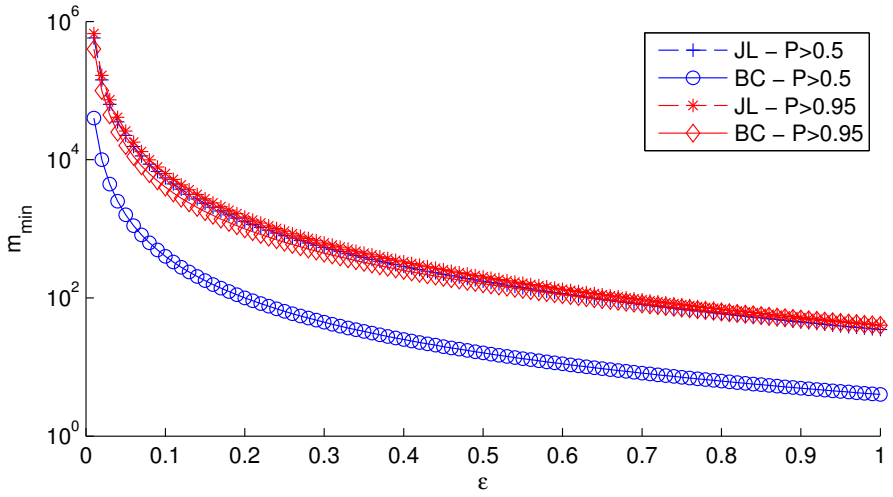

FIGURE 7.11 – Comparaison du nombre minimal m de projections nécessaires pour arriver à des probabilités supérieures à 0.5 et 0.95 en fonction de ε. On compare donc les valeurs données par les équations (7.11) et (7.7) (avec $\#Q = 1000$ pour celle-ci).

7.4 Conclusion

On a vu dans ce chapitre, par l'exemple, qu'il n'est pas nécessaire de reconstruire le signal pour travailler sur l'information contenue. L'a priori usuel "le signal se décompose de manière parcimonieuse" est remplacé par "les signaux peuvent être segmentés". Ceci permet de compresser des signaux de manière très peu coûteuse tout en gardant la possibilité de faire une segmentation après transmission. Dans le cas de signaux neuronaux où c'est souvent l'objectif, et où les ressources sont très limitées, c'est un fort avantage. Ce résultat paraît en fait très intuitif, puisque ces méthodes de segmentation reposent sur la distance entre les différents éléments et beaucoup des résultats du codage parcimonieux

3. En annexe, une autre approche utilisant la borne de Chernoff est envisagée avec de meilleurs résultats, mais les approximations faites nuisent à la fiabilité de la borne

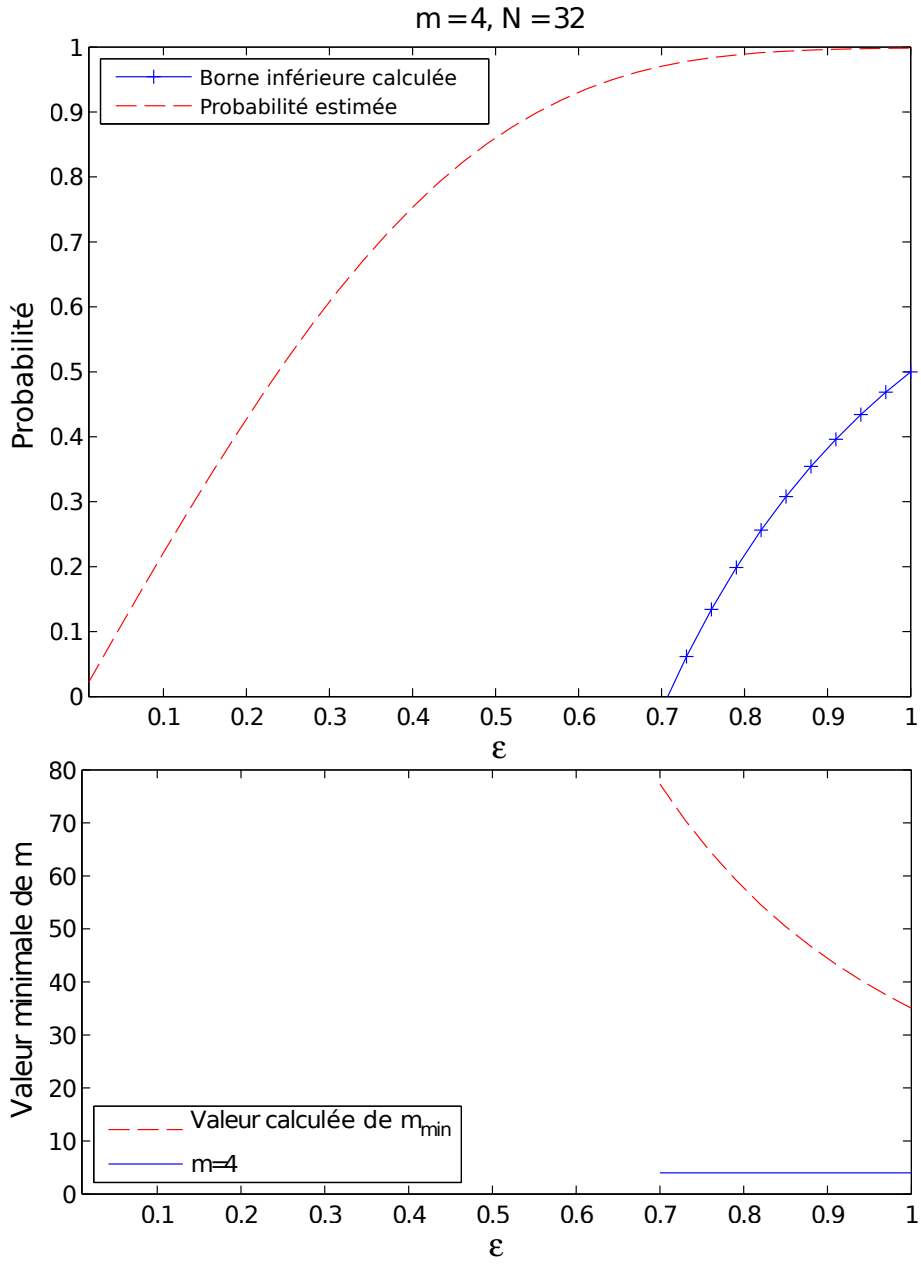

FIGURE 7.12 – Borne calculée selon (7.10) et estimation de probabilité que $\left|\frac{\|\Phi x\|}{\|x\|} - 1\right| \leq \delta$, avec $m = 4, N = 32$ (haut). Bas : valeur minimale pour assurer la même probabilité selon le lemme 1, calculée comme $m_{min} = \left(\frac{4-2\ln(1-P)}{\varepsilon^2/2+\varepsilon^3/3}\right) \ln \#Q$, où $\#Q = 1000$ et P prenant la valeur de la borne calculée sur la figure du haut. À titre de comparaison, on a tracé une ligne pour $m = 4$.

s'appuient sur la propriété d'isométrie restreinte. Cette propriété indique que les projections aléatoires ont tendance à préserver la norme des vecteurs projetés et a fortiori les distances. On propose en outre un résultat qui permet de justifier de manière statistique ce fonctionnement dans le cas de petites dimensions, ce que la littérature ne proposait pas. Ce résultat repose sur un autre qui est connu pour ne pas être très strict, cependant les

simulations montrent qu'en pratique, la projection préserve nettement mieux la norme que ce que la borne proposée suggère.

Chapitre 8

Conclusions et perspectives

Dans ces travaux de thèse, nous nous sommes intéressés aux possibilités offertes par le codage parcimonieux et plus particulièrement son utilisation pour concevoir des systèmes de compression à bas coût. En effet, la multiplication des réseaux de capteurs autonomes impose de nouvelles contraintes du point de vue énergétique et des capacités de communication. Le codage parcimonieux, dans son concept, permet de compresser un signal avec le seul a priori qu'il existe un espace de représentation dans lequel ce signal est parcimonieux, ceci à l'aide d'une simple projection, en déportant toute la complexité des calculs sur la décompression. Cela semble parfaitement adapté aux besoins d'un réseau de capteurs autonomes.

Dans cette perspective, nous avons étudié le concept du codage parcimonieux. Ce concept repose sur une hypothèse essentielle, dont il tire son nom : la parcimonie. Par cette notion, on parle de la possibilité de représenter les signaux observés avec peu de coefficients. Pour cela, on a introduit le dictionnaire de décomposition parcimonieuse Ψ. Le signal $x \in \mathbb{R}^N$ peut alors être représenté par :

$$x = \Psi\alpha = \sum_{i=1}^{N} \alpha_i \Psi_i.$$

On peut quantifier le côté parcimonieux de ce signal à l'aide de différentes normes. La plus naturelle est la norme dite ℓ_0 : c'est le nombre de coefficients différents de 0. Cependant, toutes les normes ℓ_p avec $0 < p \leq 1$ sont des indicateurs de parcimonie.

Généralement, les signaux observés dans la nature peuvent se représenter de manière parcimonieuse. C'est à partir de ce constat que de nombreuses méthodes classiques de compression sont développées, comme la compression d'image ou de musique. L'hypothèse de parcimonie pour la compression n'est donc pas nouvelle. Cependant, toutes ces méthodes nécessitent des calculs comme une transformée de Fourier discrète, ou en ondelettes. La projection utilisée en codage parcimonieux est de ce point de vue beaucoup plus avantageuse, puisque ne nécessitant que peu d'opérations. Les coefficients de représentation ne

sont pas calculés puis triés pour n'en garder qu'une partie. L'hypothèse de parcimonie intervient au moment d'inverser un système qui n'est pas inversible et qui présente une infinité de solutions : elle permet de choisir une solution parmi toutes celles possibles, en choisissant celle qui présente la norme ℓ_p $(0 \leq p \leq 1)$ la plus faible, puisque ces normes sont des indicateurs de parcimonie.

De nombreuses méthodes de décompression existent. Le chapitre 2 en a présentées quelques-unes, notamment l'optimisation convexe qui est à l'origine de l'engouement autour du codage parcimonieux. Les travaux de Candès et Donoho, entre autres, montrent l'équivalence entre la solution en norme ℓ_0 et celle en norme ℓ_1 : il y a donc une méthode qui permet d'obtenir assurément le résultat le plus parcimonieux possible, puisque la recherche de la solution de norme ℓ_0 minimale n'est pas possible. Mais les algorithmes gloutons, qui par construction donnent des solutions parcimonieuses, permettent aussi de retrouver un signal parcimonieux. Nous avons aussi retenu les méthodes itératives de moindres carrés pondérés : elles permettent d'approcher des solutions de norme ℓ_p minimale, notamment pour $p < 1$, ce qui promeut encore plus la parcimonie.

Cependant, pour pouvoir reconstruire le signal compressé, il y a plusieurs critères à respecter, concernant la matrice de codage et le dictionnaire. Tout d'abord, il est important que la matrice de codage explore toutes les colonnes du dictionnaire, pour capter de l'information quelles que soient les colonnes du dictionnaire utilisées par le signal parcimonieux. Ce critère se mesure à l'aide de la cohérence μ (p. 21) qui doit être minimale pour que la matrice soit adaptée. Ensuite, il est important que la matrice de codage se comporte presque comme une isométrie pour les signaux parcimonieux. Ce critère est caractérisé par la constante d'isométrie restreinte δ_k définie p. 35. C'est sur cette constante que s'appuient les résultats principaux à l'origine de l'engouement autour du *Compressed Sensing* : on peut définir une condition d'équivalence entre la solution de norme ℓ_0 minimale, et celle de norme ℓ_1 minimale ; ou dans le cas de signaux qui ne sont pas strictement parcimonieux, de quantifier l'erreur. Le point intéressant est que ces propriétés sont généralement vérifiées lorsque la matrice est issue d'un tirage aléatoire.

Pour mieux appréhender ces concepts, les simulations réalisées dans le chapitre 4 portent sur des exemples de signaux synthétiques, parfaitement parcimonieux dans la base canonique. Ces simulations permettent de mettre en avant deux algorithmes, parmi les quatre considérés : la programmation linéaire qui permet d'obtenir la solution de norme ℓ_1 minimale, et IRLS, qui est une méthode itérative pouvant donner la solution de norme ℓ_p minimale. Le but n'était pas de faire un comparatif exhaustif des algorithmes de reconstruction, mais d'en cerner les capacités. On constate donc qu'il est possible de reconstruire un signal parcimonieux avec un nombre restreint de mesures. De plus, ces résultats indiquent que le fait de choisir des matrices aléatoires binaires n'a pas de répercussions notables sur les capacités de codage, ce qui présente un avantage pour l'implémentation. Cependant, les expériences avec des signaux réels mettent en évidence deux choses : il est

difficile de déterminer un dictionnaire de décomposition parcimonieuse qui va permettre la recherche de solution par l'algorithme, et le fonctionnement dans un environnement imparfait, notamment avec un signal bruité, n'est pas aussi efficace qu'il ne l'est sur les exemples idéaux.

L'objectif de cette étude sur le codage parcimonieux est de mettre en avant la possibilité de réaliser de la compression à bas coût pouvant être utilisée dans des capteurs autonomes. Cette méthode de codage a été choisie parce qu'elle a l'avantage d'être très peu coûteuse en calcul. Les simulations ont montré qu'il était possible de faciliter davantage ce calcul en choisissant une matrice binaire, ce qui permet de ne plus faire de calcul sur des nombres en virgule flottante, mais de se contenter d'opérations binaires. Puisque l'objectif est de compresser des signaux relativement basses fréquences, l'échantillonnage est réalisé de manière traditionnelle, le codage s'effectuant sur le signal échantillonné et numérisé. Vouloir effectuer le codage en analogique implique trop de contraintes qui ne se justifient pas dans le cadre envisagé ici, puisqu'il n'y a pas de difficulté à échantillonner avec les technologies classiques. Il suffit donc d'implanter une matrice binaire en mémoire d'un PIC et d'effectuer les opérations d'addition et changement de signe à chaque nouvel échantillon. L'utilisation d'un simple PIC présente quelques inconvénients vis-à-vis des dimensions maximales exploitables, mais selon l'application choisie, cela peut être suffisant.

L'état de l'art et les expériences ont soulevé deux problèmes : la détermination du dictionnaire de décomposition parcimonieuse et la difficulté de reconstruction du signal. On s'est donc posé les questions suivantes :

- est-il possible de déterminer un dictionnaire de décomposition parcimonieuse par apprentissage ?
- est-il possible d'effectuer des opérations sur les signaux en se passant de l'étape de reconstruction ?

Pour la première question, nous nous sommes intéressés au problème *offline* de l'apprentissage du dictionnaire lorsque l'on dispose des signaux sous leur forme non compressée, et non pas vus au travers d'une matrice de codage. L'apprentissage du dictionnaire se fait généralement de manière itérative : il faut déterminer un dictionnaire minimisant l'erreur entre le signal et sa représentation d'un côté, et déterminer des sources parcimonieuses d'un autre côté. Cette deuxième étape est traitée au chapitre 2 puisqu'il s'agit uniquement d'une recherche de solution parcimonieuse. Plusieurs approches sont possibles pour la mise à jour du dictionnaire. Nous avons étudié deux d'entre elles : l'une s'appuie sur une méthode de gradient (Focuss-CNDL) et l'autre sur la décomposition en valeurs singulières (k-SVD). Sur des simulations, nous avons rencontré des difficultés à apprendre exactement un dictionnaire connu et notamment à régler de manière optimale les paramètres du premier algorithme. Cependant, sur un exemple concret, les deux méthodes permettent d'identifier un dictionnaire conduisant à une décomposition parcimonieuse des signaux avec une erreur contenue. Nous avons relevé des points de fonctionnement pouvant être améliorés, et permettre de sortir d'un minimum local lors de l'apprentissage, en réinitialisant certains

atomes trop corrélés avec les autres, ou trop peu utilisés.

Le deuxième point soulevé par l'étude préliminaire du codage parcimonieux est lié à la difficulté de reconstruire le signal : puisque le signal compressé contient l'information complète, est-il nécessaire de passer par l'étape coûteuse qui permet de retrouver le signal d'origine ? Ou bien est-il possible de travailler directement sur la forme compressée pour effectuer divers traitements ? Nous nous sommes appuyés sur un exemple de classification de signaux neuronaux pour constater que la projection conserve l'information sur les classes et qu'il est donc possible de faire une segmentation en se passant de la reconstruction. Ceci est dû au fait que la projection sur une matrice aléatoire préserve plus ou moins les normes des vecteurs projetés. Un résultat proposé donne une probabilité minimale d'avoir une distorsion donnée, même dans des exemples de petites dimensions. Les résultats expérimentaux ont conduit à une étude de faisabilité de composant silicium réalisant la détection et la compression des impulsions neuronales. Selon les simulations, le coût énergétique est inférieur au gain obtenu par la réduction du débit de données au niveau de la transmission.

Les travaux présentés dans ce document laissent plusieurs questions en suspens. Concernant la détermination du dictionnaire, la possibilité d'effectuer un apprentissage *online* à partir des signaux compressés reste ouverte : les méthodes étudiées dans le cas *offline* ne sont pas directement adaptables. Le traitement des données compressées n'a abordé qu'un seul problème, celui de la classification dans un environnement idéal. Il faudrait s'intéresser aux performances dans des situations plus délicates, notamment en présence de bruit et les comparer à l'état de l'art de la classification de signaux neuronaux. Mais les traitements dans le domaine compressé ne se limitent pas à la classification : la détection est une application qui devrait être possible sans reconstruction, par exemple. Enfin, un travail de conception permettant d'évaluer le coût réel d'un système de codage parcimonieux sur un capteur, et le gain apporté par son utilisation, constitue la prochaine étape de cette étude.

Annexe A

Algorithmes

A.1 Algorithmes de reconstruction

La section 2.2.2 du chapitre 2 présentent les algorithmes de reconstruction MP, OMP et IRLS. Ceux-ci sont présentés ici.

A.1.1 MP

ALGORITHME DE MATCHING PURSUIT [DDWB06]

1. On initialise le résidu $r^{(0)} = y$ et l'estimation $\hat{\Theta} = 0, \hat{\Theta} \in R^N$. On initialise le compteur d'itération $t = 1$.
2. On choisit le vecteur du dictionnaire $V = \Phi\Psi$ qui maximise la projection du résidu sur ce dictionnaire :

$$n_t = \ arg \max_{i=1,\cdots,N} \left| \frac{< r^{(t-1)}, v_i >}{\|v_i\|} \right|.$$

3. On met à jour le résidu et le coefficient de l'estimée correspondant au vecteur choisi :

$$r^{(t)} = r^{(t-1)} - \frac{< r_{t-1}, v_{n_t} >}{\|v_{n_t}\|} v_{n_t},$$

$$\hat{\Theta}_{n_t} = \hat{\Theta}_{n_t} + \frac{< r^{(t-1)}, v_{n_t} >}{\|v_{n_t}\|}.$$

4. On incrémente t. Si $t < T_{max}$ et $\|r^{(t-1)}\|_2 > \varepsilon\|y\|_2$, alors on retourne à l'étape 2 ; sinon, on continue.
5. On obtient le signal estimé $\hat{x} = \Psi\hat{\Theta}$.

A.1.2 OMP

Algorithme de Orthogonal Matching Pursuit [TG07]

1. On initialise le résidu $r_0 = y$, et l'ensemble de coefficients $\Lambda_0 = \varnothing$, l'ensemble d'atomes Θ une matrice vide . On initialise le compteur d'itération $t = 1$, le nombre maximal d'atomes k_{max} et le critère d'arrêt ε. [a]
2. On choisit le vecteur du dictionnaire qui maximise la projection du résidu sur $V = \Phi\Psi$:
$$n_t = \ arg \max_{i=1,\cdots,N} \left| \frac{< r_{t-1}, v_i >}{\|v_i\|} \right|.$$
3. On met à jour $\Lambda^{(t)}$ et $\Theta^{(t)}$:
$$\Lambda^{(t)} = \Lambda^{(t-1)} \cup n_t \,;$$
$$\Theta^{(t)} = [\Theta^{(t-1)} \, V_{n_t}].$$
4. On résout le problème des moindres carrés suivant :
$$a^{(t)} = \arg\min_x \|y - \Theta^{(t)} x\|_2.$$
5. On calcule la nouvelle estimation α_t et le résidu :
$$\alpha^{(t)} = \Theta^{(t)} a^{(t)},$$
$$r^{(t)} = y - \alpha^{(t)}.$$
6. On incrémente t. Si $t < k_{max}$ et $\|r^{(t)}\|_2 > \varepsilon \|y\|_2$, alors on retourne à l'étape 2 ; sinon, on passe à la suite.
7. On obtient le signal estimé $\hat{x} = \alpha_t$.

a. On utilisera $\varepsilon = 10^{-3}$.

A.1.3 IRLS

ALGORITHME IRLS [CY08]

1. On initialise le compteur d'itération $c = 1$, le nombre maximum d'itérations $c_{\max}$, le coefficient de régularisation à $\varepsilon = 0.1$, et la solution de départ $x^{(0)} = \Phi^T(\Phi\Phi^T)^{-1}y$
2. On calcule les poids $w_i = \left(|x_i^{(c-1)}|^2 + \varepsilon\right)^{p/2-1}$;
3. On calcule la prochaine itération $x^{(c)} = Q_n\Phi^T(\Phi Q_n\Phi^T)^{-1}y$;
4. Si $|\|x^{(c)}\|_2 - \|x^{(c-1)}\|_2| > \sqrt{\varepsilon}/100$ et $c < C_{max}$, on retourne à l'étape 2 après avoir incrémenté le compteur c ;
5. Si $|\|x^{(c)}\|_2 - \|x^{(c-1)}\|_2| < \sqrt{\varepsilon}/100$ et $\varepsilon > 10^{-8}$ alors on modifie $\varepsilon = \varepsilon/10$ et si $c < c_{\max}$, on incrémente c et on retourne à l'étape 2 ; sinon on termine.

Annexe B

Implémentation Matlab

Les simulations sont mises en œuvre avec Matlab. Dans le cas général, la longueur du signal est fixée à $N = 256$. Les signaux x sont générés aléatoirement : la position des pics est déterminée à l'aide de la fonction *randperm* et leurs amplitudes sont aléatoires, indépendantes, selon une loi normale $N(0, 1)$.

On observe ce signal via une matrice dont les éléments sont i.i.d. selon une loi normale centrée puis les lignes sont orthonormalisées. Cette matrice est de dimension $m \times N$.

On utilise aussi une version sur 1 bit ($\pm\frac{1}{\sqrt{N}}$) pour observer l'effet des matrices binaires. L'effet certain est que l'orthogonalité entre les lignes est perdue. Cependant, on reste dans une situation proche de l'orthogonalité, les lignes sont faiblement corrélées entre elles.

Pour les matrices Walsh et DFT, une permutation aléatoire des indices des lignes est effectuée et on en prend les m premières.

L'observation $y = Ax$ est ensuite reconstruite par les algorithmes, et on mesure à chaque fois l'erreur relative $\varepsilon = \frac{\|\hat{x}-x\|_2}{\|x\|_2}$.

Pour mesurer la performance en reconstruction, on détermine qu'un signal est reconstruit parfaitement si $\varepsilon < 10^{-3}$.

B.1 Génération des signaux de test

Les signaux sont générés de façon aléatoire de la manière suivante :

```
x = zeros(N,1);
q = randperm(N);
x(q(1:k)) = randn(k,1);
```

La dernière ligne peut éventuellement être remplacée par :

```
x(q(1:k)) = sign(randn(k,1));
```

ou par :

```
x(q(1:k)) = ones(k,1);
```

B.2 Génération des matrices

Les matrices aléatoires gaussiennes sont générées de la manière suivante :

```
A = orth(randn(m,N)')';
```

Les matrices de Walsh-Hadamard sont générées avec la fonction suivante :

```
function H = walshmtx(mtxsize)

n_iter = round(log(mtxsize)/log(2));
H = 1;
for iter=1:n_iter
    H = [H H; H -H];
end
```

Ensuite, on sélectionne des lignes au hasard :

```
W = walshmtx(N);
q = randperm(N);
A = W(q(1:m),:);
```

De manière identique, en remplaçant la fonction *walshmtx* par la fonction *dftmtx* de Matlab, on génère les matrices de Fourier.

B.3 Algorithmes de reconstruction

Les algorithmes, décrits par l'annexe A, sont programmés avec le langage Matlab et retranscrits ici.

B.3.1 IRLS

```
function xest = irls(y,Phi,p,MAXITER)
% premiere estimation : solution moindres carres
x = Phi' * 1/(Phi * Phi') * y;
eps = 0.1;
for k=1:MAXITER
    w = abs(x.^2 + eps).^(p/2 - 1);
    Q = diag(1./w);
    x_0 = x;
    x = Q * Phi' * ((Phi * Q * Phi') \ y);
    if (abs(norm(x_0)-norm(x))<sqrt(eps)/100)
        if (eps>1e-8)
            eps = eps/10;
        end
    end
end
xest = x;
```

B.3.2 Basis Pursuit

On fait appel à la fonction *l1eq_pd* de la bibliothèque ℓ_1-MAGIC [1] avec les paramètres par défaut.

B.3.3 Matching Pursuit

```
function xest = mp(y,Phi,Psi,maxiter)
N = length(Psi);
thetaest=zeros(N,1);
rt=y;
iter=1;
error=1;
V = Phi * Psi;
v_norm = diag(V'*V)';
while (iter<=maxiter && error>1e-10);
    Proj = rt'*V ./ v_norm;
    [c,v]=max(abs(Proj));
    nt=V(:,v);
    rt=rt-Proj(v)*nt;
    thetaest(v)=thetaest(v)+Proj(v);
    error=(rt'*rt)/(y'*y);
    iter=iter+1;
end
xest=Psi*thetaest;
```

B.4 Erreur de reconstruction

Il est nécessaire d'établir un critère pour déterminer si une reconstruction $\hat{x}$ est exacte ou non. Un moyen évident est de mesurer la norme relative de l'erreur en norme ℓ_2 $\frac{\|\hat{x}-x\|_2}{\|x\|_2}$ et de fixer un seuil en dessous duquel la reconstruction est considérée comme exacte. Nous avons fixé le seuil à 10^{-3}. Les résultats de la figure B.1 montrent que le critère de sélection pour déterminer si un signal est bien reconstruit, choisi à 10^{-3} est suffisant : en effet, dans le cas des signaux bien reconstruits, c'est-à-dire dont l'erreur relative est inférieure au seuil de 10^{-3}, l'erreur de reconstruction observée se situe principalement autour de 10^{-7}.

B.5 Apprentissage de dictionnaire

Les algorithmes d'apprentissage de dictionnaire présentés dans le chapitre 6 sont programmés en langage Matlab et retranscrits ci-dessous.

B.5.1 Focuss-CNDL

1. Disponible à l'adresse `http://www.l1-magic.org`

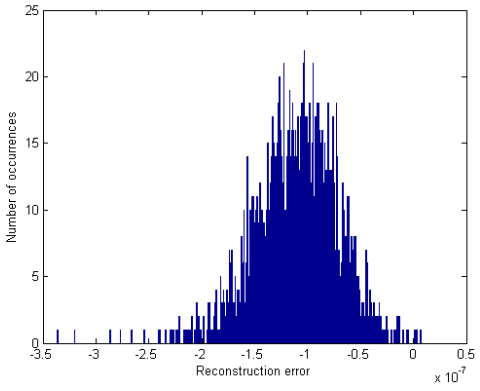

FIGURE B.1 – Histogramme des erreurs relatives en norme ℓ_2 de reconstruction par l'algorithme IRLS avec $N = 256$, $M = 40$, $k = 5$ et $p = 0.1$ lorsque l'erreur est inférieure à 10^{-3}

```
function [Aest, Xest] =focuss_cndl(Y,k,N,p,gamma,lambda_max,nb_paquet,nb_update,maxiter)

[m,nbtr]=size(Y);
update = zeros(m,N);

Aest = Y(:,1:N);
Xest = Aest'*((Aest*Aest')\Y);

for iter=1:maxiter
    idx = 0;
    for ipaq = 1:nb_paquet
        for j=1:nb_update
            idx = idx+1;

            lambda = max(lambda_max*(1-norm(Y(:,idx)-Aest*Xest(:,idx))/norm(Y(:,idx))),lambda_max);

            Qf = diag(abs(Xest(:,idx)).^(2-p));
            Xest(:,idx) = Qf*Aest'*(((lambda*norm(Xest(:,idx),p)^(1-p)*eye(m))+Aest*Qf*Aest')\y);

            if (length(find(Xest(:,mod(idx,nbtr)+1)>1e-6))<k-2)
                Xest(:,mod(idx,nbtr)+1)=randn(N,1);
            end
        end

        Xs = Xest(:,((ipaq-1)*nb_update+1):(ipaq*nb_update));
        [a,b] = sort(abs(Xs),'descend');
        Xs = x.*(b<=k);

        Syx= Y(:,((ipaq-1)*nb_update+1):(ipaq*nb_update))*Xs' / (nb_update);
        Sxx= Xs*Xs' / (nb_update);

        dAest = Aest*Sxx - Syx;
        for j=1:N
            update(:,j) = gamma *(eye(m) - N*Aest(:,j)*Aest(:,j)')*dAest(:,j);
        end
        Aest = Aest -update;
```

```
        for j=1:N
            a=(sqrt(N)*norm(Aest(:,j)));
            Aest(:,j) = Aest(:,j)/a;
            Xest(j,:) = Xest(j,:)*a;
        end
    end


    c = sign(Aest(1,:));
    Aest = Aest.*(ones(m,1)*sign(Aest(1,:)));
    [a,b] = sort(Aest(1,:));
    Aest = Aest(:,b);
    Xest = Xest(b,:);
    Xest = Xest.*(c'*ones(1,nbtr));

end
```

B.5.2 k-SVD

```
function D = ksvd(Y,N,D0)
[m,n] =size(Y);
if nargin <3
     D0 = Y(:,1:N);
     D0 = D0/sqrt(trace(D0*D0'));;
end
X=zeros(N,n);
maxiter = 500;
maxk = round(N/10);
IdN=eye(N);
for i_iter = 1:maxiter
   for i_vec=1:n;
       X(:,i_vec) = omp(Y(:,i_vec),D0,IdN,maxk);
   end
   D=D0;
   for i_col=1:N
       idx_nz = find(X(i_col,:));
       if(length(idx_nz)>0)
       Ek= Y - D*X + D(:,i_col)*X(i_col,:);
       Omega = zeros(n,length(idx_nz));
       for inz=1:length(idx_nz)
           Omega(idx_nz(inz),inz)=1;
       end
       EkR=Ek*Omega;
       [U,S,V] = svd(EkR);

       D0(:,i_col)=U(:,1);
       X(i_col,idx_nz)=S(1,1)*V(:,1)';
       end
   end
end
```

Annexe C

Implémentation analogique du codeur parcimonieux.

Dans le chapitre 5, on présente une implémentation numérique du codage parcimonieux. Cependant, avant de faire ce choix, nous avons aussi envisagé une implémentation analogique puisque celle-ci existe dans la littérature [LKD+07, KLW+07]. L'étude de cette possibilité, que nous n'avons pas retenue, est présentée ici.

C.1 Convertisseur analogique-information

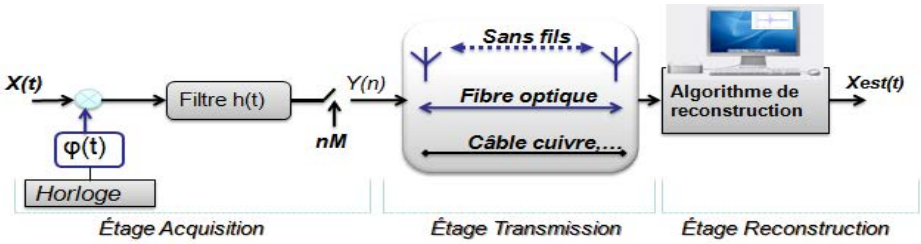

FIGURE C.1 – Convertisseur Analogique-Information

Dans cette partie, nous abordons le fonctionnement du convertisseur analogique-information. Dans les exemples du chapitre 2, le problème est traité sous une forme entièrement analytique : l'observation du signal s'écrivant

$$y = \Phi x.$$

Cette forme suppose que le signal x est déjà échantillonné, en respectant les conditions de Shannon. Avec le convertisseur analogique-information, il n'y a pas d'échantillonnage respectant le critère de Shannon avant d'appliquer une projection sur une matrice d'observation.

Dans le CAI, l'opération de projection, c'est-à-dire un produit scalaire, se fait sur des signaux continus, $x(t)$ et $\phi_i(t)$. Alors qu'en discret, le produit scalaire s'écrit

$$< x, y >= \sum_{k=1}^{N} x_k y_k,$$

le produit scalaire usuel pour les fonctions de carré sommable est

$$< x(t), \phi_i(t) >= \int_{\Omega} x(t)\phi_i(t)dt,$$

où Ω est le domaine de définition des signaux. Dans notre cas, on veut que ce domaine corresponde à la durée T_e d'une période de la séquence $\psi_i(t)$. À un instant nT_e donné, l'observation $y(nT_e)$ est un vecteur de m éléments obtenus de la manière suivante :

$$y_i(nT_e) = \int_{(n-1)T_e}^{nT_e} \phi_i(t)x(t)dt, \text{ pour } 1 \leq i \leq m$$

C'est ce résultat qui est échantillonné à la fréquence $F_e = \frac{1}{T_e}$, inférieure à la fréquence requise par le théorème de Shannon. L'intégrale est réalisée à l'aide d'un filtre intégrateur, dont on décharge le condensateur à chaque fin de période T_e après l'échantillonnage.

La figure C.2 décrit le système en détail.

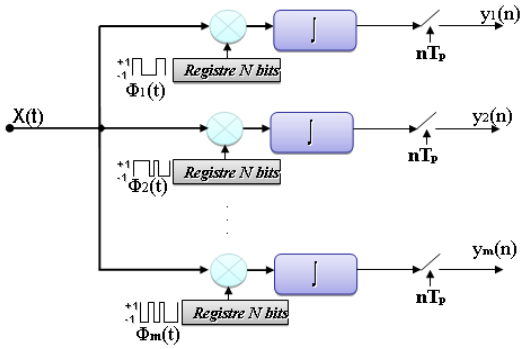

FIGURE C.2 – Principe du convertisseur analogique-information

Le convertisseur analogique-information se compose de quatre éléments principaux, qui sont discutés en détail dans la suite de ce chapitre :

- Une méthode de génération des signaux ϕ_j ;
- Le multiplieur, pour mélanger les deux signaux analogiques ;
- Un intégrateur ;
- Un convertisseur analogique-numérique.

C.1.1 La génération des séquences de mixage ϕ_i

Pour des raisons pratiques, on a choisi d'utiliser des séquences de projection binaires, constituées de valeurs ± 1 : il n'est alors plus nécessaire de multiplier deux signaux analogiques, mais seulement de changer le signe du signal. Le stockage est de plus facilité, il suffit d'un bit, et la séquence $\phi_i(t)$ est plus facile à générer (ou n'a pas besoin d'être générée explicitement s'il s'agit seulement d'une commande d'un circuit inverseur.) Pour mémoriser et restaurer ces signaux, nous utiliserons des registres à décalage permettant le stockage de la chaine pseudo-aléatoire et sa restitution de manière périodique, à une fréquence T_e déterminée par l'horloge. Plus exactement, l'horloge fonctionnera à la fréquence $N \times F_e$ où N est la longueur de la chaine. Dans la pratique, on préfère utiliser un circuit intégré réalisant cette fonction, implémenté sur une technologie CMOS (Ex : fig. C.3) pour une consommation électrique réduite. De plus, on peut profiter des tensions bipolaires $\pm Vdd$ pour représenter les deux états possibles des éléments de la matrice binaire.

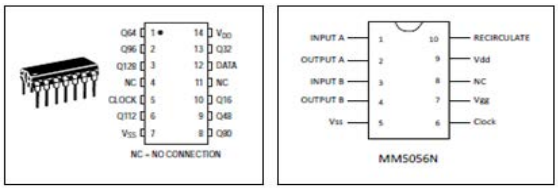

FIGURE C.3 – À gauche, le circuit MC14562B qui réalise un registre à décalage de 128 bits (cas $N = 128$.) À droite, le registre à décalage MM5056 de 2×256 bits ($N = 256$ ou $N = 512$).

Dans le cas où N serait plus grand, il est possible de monter les registres en cascade.

C.1.2 Le multiplieur

Il existe de nombreuses solutions permettant de multiplier deux signaux analogiques. Une liste non-exhaustive est présentée ci-dessous.

C.1.2.1 Un modulateur en anneaux

C'est un circuit très utilisé en télécommunication pour la modulation des signaux, son principe de fonctionnement est relativement simple avec l'aide de 4 diodes (fig. C.4.) L'utilisation de diodes, et notamment leur résistance interne, pose problème. De plus, ce circuit n'est pas miniaturisable [Dev] [1]. De plus, on veut seulement changer le signe du signal, ce qui ne nécessite pas un tel montage. Par conséquent, la solution du modulateur en anneau n'est pas retenue.

1. http://www.analog.com/static/imported-files/tutorials/MT-080.pdf

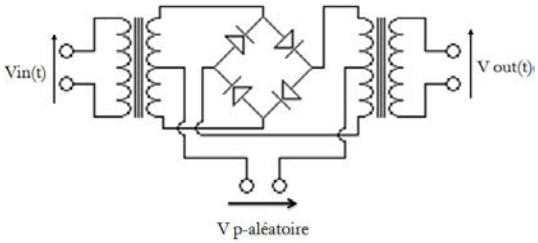

FIGURE C.4 – *Modulateur en anneaux (image Wikipedia)*

C.1.2.2 Un modulateur 4 quadrants

Ce circuit, appelé aussi *cellule de Gilbert* [Gil68], est la solution retenue par [LKD+07], dont le principe est représenté sur la figure C.5. Ce type de modulateur est également très utilisé pour mixer des signaux ultra-large bandes [TW04]. Le composant très répandu AD633 [2] réalise cette fonction. Sa bande-passante est de 1Mhz, ce qui est largement suffisant pour nos signaux dont la bande est inférieure à 10 kHz. Sa consommation maximale est de 60 mW (15V/4mA.) De plus, le prix n'est pas négligeable, puisqu'il sera multiplié m fois, malgré l'appellation *low cost* du composant : environ 4$.

Enfin, ce modulateur permet de multiplier deux signaux analogiques alors qu'il suffit, puisqu'on utilise des matrices binaires, de changer le signe du signal.

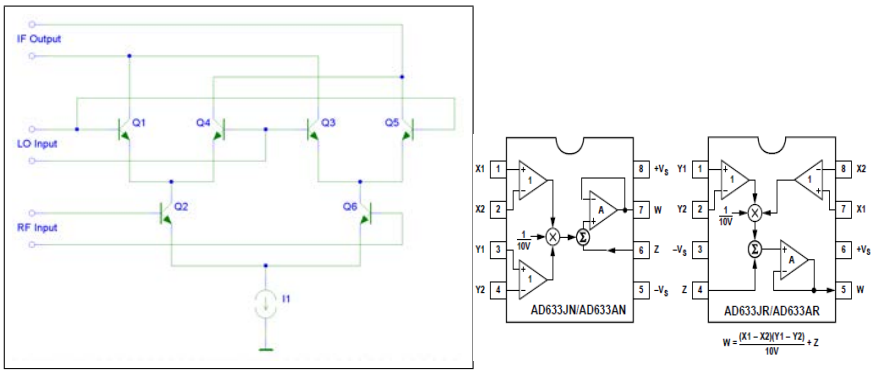

FIGURE C.5 – *Cellule de Gilbert et schéma du circuit intégré AD633 réalisant la fonction de modulateur 4 quadrants.*

C.1.2.3 Montage d'amplificateurs opérationnels

Étant donné que les fréquences de signaux envisagées n'excèdent pas 50 kHz, il est envisageable d'utiliser des amplificateurs opérationnels dans un montage au principe simple (fig.C.6,) qui ne fait que moduler le signe du signal en fonction de la séquence binaire de projection. Dans un premier temps, la séquence pseudo-aléatoire issue du registre à décalage

2. `http://www.analog.com/static/imported-files/data_sheets/AD633.pdf`

module la base (ou la grille) de deux transistors de façon à ce que l'un soit passant quand l'autre est bloqué. Les AOP correspondants transmettent le signal lorsque leur transistor est passant, l'un quand la séquence est positive, l'autre quand la séquence est négative. Enfin, le dernier AOP permet d'additionner les deux tensions obtenues en prenant soin d'inverser celle qui correspond au éléments négatifs de la séquence. Finalement, ce circuit produit un signal $m(t)$ qui correspond à la séquence pseudo-aléatoire modulée par $x(t)$.

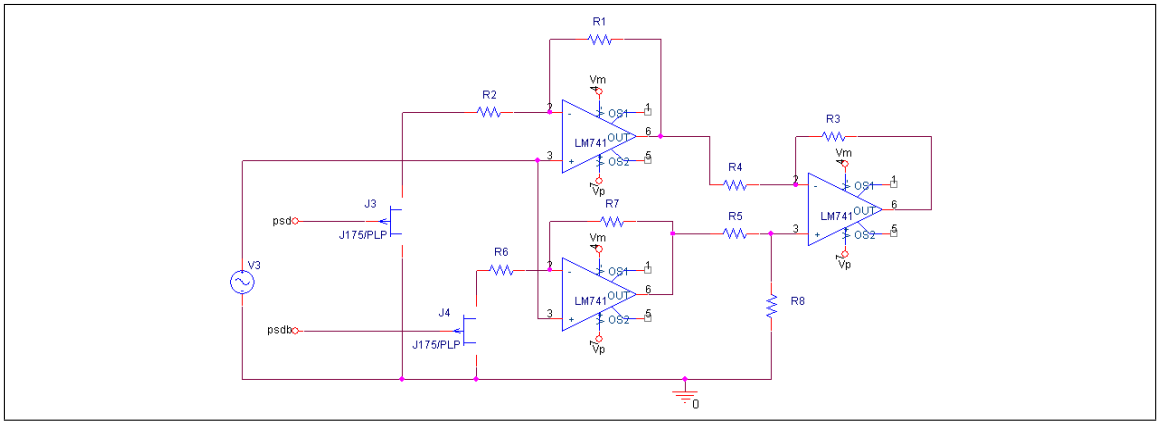

FIGURE C.6 – Montage d'amplificateurs opérationnels pour la multiplication du signal par des séquences pseudo-aléatoires binaires ±1. Les deux AOP de gauche sont commandés par deux transistors. Ces transistors sont alternativement passants ou ouverts selon la valeur de la séquence ϕ_i, de manière symétrique : quand l'un est passant, l'autre est ouvert. L'AOP de droite permet d'inverser le signal lorsque l'AOP du bas transmet le signal. Ainsi, le signal est modulé en fonction de la séquence phi_i

La consommation de certains AOP est relativement élevée : le LM741 (fig. C.6,) affiche une consommation nominale de 50 mW. Sachant qu'il y en a 3 dans le montage, la consommation est potentiellement plus élevée qu'avec le multiplieur 4 quadrants AD633. Par contre, le coût est de l'ordre de la dizaine de centimes d'euro.

C.1.2.4 Montage avec OTA (Operational Transconductance Amplifier)

Le montage précédent a inspiré un autre montage (figure C.7) fait à partir d'un amplificateur de transconductance à sortie différentielle, le but étant simplement de sélectionner à la sortie de l'OTA soit la tension $+V_{in}$ ou $-V_{in}$. Cette sélection est faite à la cadence de la séquence pseudo-aléatoire, ainsi on obtient les mêmes résultats que précédemment.

L'amplificateur MAX435 est donné pour une consommation nominale de 350 mW et un coût d'environ 0.5 €.

C.1.2.5 Choix

On retiendra donc le montage avec AOP, car coûtant moins cher que le multiplieur AD633, et consommant moins que le montage avec OTA.

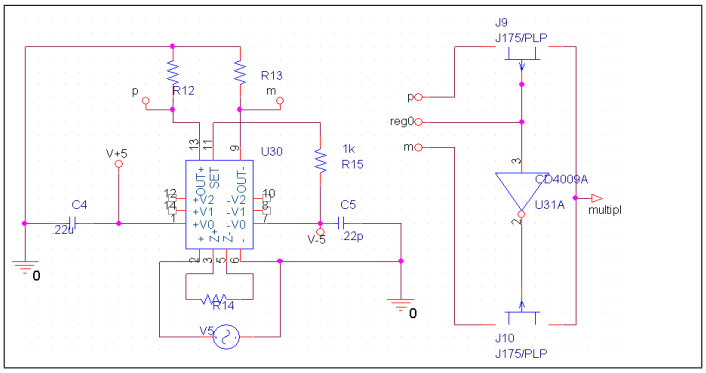

FIGURE C.7 – Design du multiplieur avec un OTA (MAX435)

C.1.3 Le montage intégrateur

Afin de réaliser un produit scalaire, il est nécessaire de sommer le résultat de la multiplication. La solution la plus simple est d'utiliser un filtre RC (passe-bas) dont la tension de sortie est équivalente à l'intégrale de la tension d'entrée lorsque les fréquences dépassent largement la fréquence de coupure f_c du filtre.

Cependant, un montage passif simple n'est pas adapté à l'application. En effet, l'intégrale s'effectue sur une durée finie, correspondant à la durée de la séquence de projection : il est nécessaire de remettre à zéro l'accumulateur à chaque période de la séquence pseudo-aléatoire. Pour effectuer cette remise à zéro, une horloge commande un circuit de décharge du condensateur à la fréquence F_e, c'est-à-dire tous les N changements d'état du registre. Ce circuit est constitué d'un transistor et d'une résistance ($R9$ sur la figure C.8).

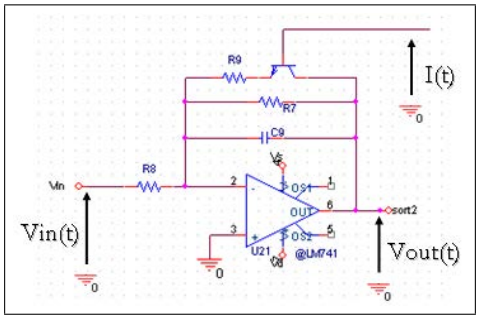

FIGURE C.8 – Filtre RC avec un AOP et circuit de décharge.

Ici, $f_c = \frac{1}{2\pi R_8 C}$ et dans la bande passante du filtre $f_s < f_c$: $V_{out} = -\frac{R_7}{R_8} V_{in}$. Mais pour $f_s >> f_c$:

$$V_{out}(t) = -\frac{1}{R_7 C} \int_{nT_p - T_p}^{nT_p} V_{in}(\tau) d\tau \, . \tag{C.1}$$

C.1.4 L'échantillonneur basse fréquence et le convertisseur analogique-numérique

Après l'intégration, il suffit de récupérer de manière périodique (tous les T_e : période des séquences pseudo-aléatoires) les échantillons du signal $y(t)$ obtenu en sortie du filtre intégrateur. Cela nous permettra d'aboutir à l'expression ci-dessous C.2 qui représente l'équivalent analogique des différents coefficients numériques de y, $y_j = \sum_i^N \phi_{ij} x_i$ définis dans la multiplication matricielle.

$$Vout(t) \;=\; \frac{1}{RC} \int_{nT_p - T_p}^{nT_p} x(t)\phi_i(t)dt. \tag{C.2}$$

Pour effectuer cette tâche, on utilisera un simple échantillonneur-bloqueur d'ordre 0 qui récupère la valeur de y et la maintient pendant un instant T_c. Dans le but de numériser le signal en vue d'une transmission ultérieure, on utilise un convertisseur analogique numérique (CAN) usuel qui fonctionne à la fréquence *basse* F_e.

C.1.5 Conclusions

On propose ici une implémentation du convertisseur analogique-information : le système se compose de m branches effectuant une projection du signal sur une séquence de projection, représentation continue d'une ligne de la matrice d'observation Φ. Pour cela, il faut un registre enregistrant les états binaires de la séquence, un multiplicateur pour changer le signe du signal en fonction de ces états, un filtre intégrateur, avec une fonction de remise à zéro pour sommer le résultat de la multiplication, et enfin, un convertisseur analogique-numérique *basse* fréquence. Bien que cette solution soit envisageable et flexible, il faut remarquer que lorsqu'on augmente le nombre de projections, on multiplie les branches du circuit (l'ensemble registre-multiplieur-intégrateur-CAN.) Il est éventuellement possible de réduire le nombre de convertisseurs analogiques-numériques, en utilisant un unique CAN travaillant à $m \times F_e$ et m échantillonneurs-bloqueurs pour introduire des retards, et convertir la valeur de chaque branche de manière séquentielle. Dans les deux cas, on multiplie les composants avec le nombre de projections.

Les fréquences envisagées dans notre application, inférieures à 5 kHz, ne justifient pas les inconvénients induits par la solution présentée ici, notamment la consommation élevée et le coût. Cependant, l'approche CAI, telle que décrite ici, reste valide et peut être utilisée : on lui préférera cependant les applications où les solutions numériques ne sont plus possibles, ou trop coûteuses, comme l'ultra-large bande.

Annexe D

Méthodes numériques pour le calcul des angles entre des sous espaces linéaires [BG73]

Dans le chapitre 6, nous mentionnons la possibilité de mesurer les angles entre sous-espaces afin de comparer le dictionnaire appris au dictionnaire d'origine. Le principe en est présenté dans cette annexe.

Soit F et G deux sous-espaces linéaires appartenant à un espace unitaire E^m où :

$$p = dim(F) \geq dim(G) = q \geq 1.$$

Le plus petit angle entre F et G, noté $\theta_1(F,G) = \theta_1 \in [0, 2\pi]$, est défini par :

$$\cos\theta_1 = \underset{u\in F}{max}\ \underset{v\in G}{max}\ u^H v, \quad \|u\|_2 = 1,\ \|v\|_2 = 1. \tag{D.1}$$

On suppose que le maximum est atteint pour $u = u_1$ et $v = v_1$. L'angle $\theta_2(F,G)$ est défini comme le plus petit angle entre le complément orthogonal à F par rapport à u_1 (noté $[F^{\perp}]^{-}$), et celui de G par rapport à v_1 (noté $[g^{\perp}]^{-}$). On continue de cette manière jusqu'à ce que l'un des deux sous-espaces devienne vide. Ceci nous mène à la définition suivante :

Définition 12 *Les angles principaux $\theta_k \in [0, 2\pi]$ entre F et G sont récursivement définis pour $k = 1, 2..., q$ de la manière suivante :*

$$\cos\theta_k = \underset{u\in F}{max}\ \underset{v\in G}{max}\ u^H v = u_k^H v_k, \quad \|u\|_2 = 1,\ \|v\|_2 = 1$$

où $u_j^H u = 0, \quad v_j^H v = 0, \quad j = 1, 2, ..., k-1$

Les vecteurs $U = (u_1, u_2, ..., u_g)$ et $V = (v_1, v_2, ..., v_g)$ sont les vecteurs principaux de F et G respectivement.

Pour une matrice A, on note l'espace image $Im(A)$ et l'espace des noyaux $N(A)$ avec :

$$Im(A) = \{u|Ax = u\}, \quad \text{Ker}(A) = \{x|Ax = 0\}$$

Soient : $F = Im(A)$ et $G = Im(B)$, où A et B sont deux matrices rectangulaires. Les corrélations canoniques sont ainsi égales à $\cos\theta_k$. On a de plus :

$$\cos\theta_k = \sigma_k, \quad u_k = Ay_k, \quad v_k = Bz_k, \quad k = 1, 2, ..., q,$$

où les $\sigma_k \geq 0$ sont les valeurs propres et y_k et z_k les vecteurs propres relatifs au problème des valeurs propres généralisé :

$$\begin{pmatrix} 0 & A^H B \\ B^H A & 0 \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \sigma \begin{pmatrix} A^H A & 0 \\ 0 & B^H B \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix}$$

Plusieurs méthodes ont été proposées pour le calcul numérique de la corrélation canonique dont les algorithmes de décomposition en valeurs singulières.

D.1 Résolution du problème des valeurs propres généralisé en utilisant la décomposition en valeurs singulières

Théorème 12 *Soient A et B deux matrices de deux bases unitaires relatives à deux sous-espaces d'un espace unitaire E^m. Soit : $M = A^H B$.*

La décomposition en valeurs singulières de cette matrice $(p \times q)$ est la suivante :

$$M = YCZ^H, \quad C = diag(\sigma_1, \sigma_2, ..., \sigma_q), \quad Y^H Y = Z^H Z = ZZ^H = I_q \tag{D.2}$$

On suppose que $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_q$. Les angles principaux θ_k et les vecteurs principaux qui leurs sont associés sont donnés par :

$$\cos\theta_k = \sigma_k(M), \quad U = AY, \quad V = BZ \tag{D.3}$$

D.1.0.0.1 Démonstration : Les valeurs singulières et les vecteurs singuliers d'une matrice M sont caractérisés par :

$$\sigma_k = \max_{\|y\|_2 = \|z\|_2 = 1} (y^H M z) = y_k^H M z_k, \quad avec : \; y^H y_j = z^H z_j = 0, \quad j = 1, ..., k-1 \tag{D.4}$$

Si on pose $u = Ay \in Im(A)$ et $v = Bz \in Im(B)$, ceci implique :

$$\|u\|_2 = \|y\|_2, \; \|v\|_2 = \|z\|_2 \quad \text{donc} \quad y^H y_j = u^H u_j, \; z^H z_j = v^H v_j$$

Puisque $y^H M z = y^H A B z = u^H v$, l'équation (D.4) devient donc :

$$\sigma_k = \underset{\|y\|_2=\|z\|_2=1}{max} (u^H v) = u_k^H v_k \quad avec \quad u^H u_j = v^H v_j = 0, \quad j = 1, ..., k-1$$

L'équation (D.3) découle directement de la définition des angles et des vecteurs principaux (D.1), ce qui conclut la démonstration.

Annexe E

Conservation de la norme par projection sur une matrice binaire

Le résultat de la proposition 2 du chapitre 7 est obtenu à l'aide de l'inégalité de Bienaymé-Chebychev. Cette inégalité est connue pour être assez lâche. [PL11] calcule la propriété d'isométrie restreinte pour les matrices aléatoires gaussiennes à l'aide de la borne de Chernoff. Le calcul suivant s'inspire des résultats présentés dans [PL11] pour approfondir le résultat de la proposition 2 page 91.

D'après l'inégalité de Chernoff, on a :

$$P(\|\Phi x\|^2 \geq \|x\|^2(1+\delta)) \leq \mathbb{E}\left\{e^{t\|\Phi x\|^2 - t\|x\|^2(1+\delta)}\right\} = \mathbb{E}\left\{e^{t\|\Phi x\|^2}\right\} e^{-t\|x\|^2(1+\delta)}, \text{ pour } t > 0. \tag{E.1}$$

On calcule donc le terme d'espérance :

$$\mathbb{E}\left\{e^{t\|\Phi x\|^2}\right\} = \mathbb{E}\left\{e^{t\sum_{i=1}^{m} y_i^2}\right\} = \mathbb{E}\left\{\prod_{i=1}^{m} e^{ty_i^2}\right\} = \prod_{i=1}^{m} \mathbb{E}\left\{e^{ty_i^2}\right\}, \tag{E.2}$$

car les y_i sont indépendants les uns des autres. On va donc calculer le terme suivant :

$$\mathbb{E}\left\{e^{ty_i^2}\right\} = \mathbb{E}\left\{e^{t\left(\sum_{j=1}^{N}\phi_{i,j}x_j\right)^2}\right\} = \mathbb{E}\left\{e^{t\left(\sum_{j=1}^{N}\phi_{i,j}^2 x_j^2 + \sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}\phi_{i,k}\phi_{i,l}x_k x_l\right)}\right\} \tag{E.3}$$

$$= \mathbb{E}\left\{ e^{t\left(\frac{1}{m}\sum_{j=1}^{N} x_j^2 + \sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N} \phi_{i,k}\phi_{i,l}x_k x_l\right)} \right\} = e^{\frac{t\|x\|^2}{m}} \mathbb{E}\left\{ e^{t\left(\sum_{\substack{k,l=1\\k\neq l}}^{N} \phi_{i,k}\phi_{i,l}x_k x_l\right)} \right\} \tag{E.4}$$

$$= e^{\frac{t\|x\|^2}{m}} \mathbb{E}\left\{ \prod_{\substack{k,l=1\\k\neq l}}^{N} e^{t\left(\phi_{i,k}\phi_{i,l}x_k x_l\right)} \right\} = e^{\frac{t\|x\|^2}{m}} \prod_{\substack{k,l=1\\k\neq l}}^{N} \mathbb{E}\left\{ e^{t\left(\phi_{i,k}\phi_{i,l}x_k x_l\right)} \right\} \tag{E.5}$$

car $\phi_{i,k}\phi_{i,l}$ et $\phi_{i,k}\phi_{i,l'}$ sont indépendants si $l \neq l'$, et comme $P(\phi_{i,j}\phi_{i,l} = \frac{1}{m})) = \frac{1}{2} = P(\phi_{i,j}\phi_{i,l} = -\frac{1}{m})$, on a alors :

$$\mathbb{E}\left\{ e^{ty_i^2} \right\} = e^{\frac{t\|x\|^2}{m}} \prod_{\substack{k,l=1\\k\neq l}}^{N} \left(\frac{e^{\frac{t}{m}x_k x_l} + e^{-\frac{t}{m}x_k x_l}}{2} \right) \tag{E.6}$$

et par conséquent :

$$\mathbb{E}\left\{ e^{t\|\Phi x\|^2} \right\} = e^{t\|x\|^2} \prod_{\substack{k,l=1\\k\neq l}}^{N} \left(\frac{e^{\frac{t}{m}x_k x_l} + e^{-\frac{t}{m}x_k x_l}}{2} \right)^m . \tag{E.7}$$

On en déduit donc le résultat suivant :

$$P(\|\Phi x\|^2 \geq \|x\|^2(1+\delta)) \leq e^{-t\delta\|x\|^2} \prod_{\substack{k,l=1\\k\neq l}}^{N} \left(\frac{e^{\frac{t}{m}x_k x_l} + e^{-\frac{t}{m}x_k x_l}}{2} \right)^m = f(t). \tag{E.8}$$

Il faut maintenant trouver le t qui minimise $f(t)$. Comme le logarithme est une fonction monotone croissante, cela revient à trouver le t minimisant $\ln f(t)$:

$$\frac{\partial \ln f(t)}{\partial t} = \frac{\partial}{\partial t}\left[-t\delta\|x\|^2 + m\sum_{\substack{k,l=1\\k\neq l}}^{N} \ln\left(\frac{e^{\frac{t}{m}x_k x_l} + e^{-\frac{t}{m}x_k x_l}}{2} \right) \right] \tag{E.9}$$

$$= -\delta\|x\|^2 + m\sum_{\substack{k,l=1\\k\neq l}}^{N} \frac{x_k x_l}{m} \left(\frac{e^{\frac{t}{m}x_k x_l} - e^{-\frac{t}{m}x_k x_l}}{e^{\frac{t}{m}x_k x_l} + e^{-\frac{t}{m}x_k x_l}} \right) \tag{E.10}$$

$$= -\delta\|x\|^2 + \sum_{\substack{k,l=1\\k\neq l}}^{N} x_k x_l \tanh\left(\frac{t}{m}x_k x_l \right). \tag{E.11}$$

On cherche donc $t > 0$ tel que :

$$0 = -\delta\|x\|^2 + \sum_{\substack{k,l=1 \\ k\neq l}}^{N} x_k x_l \tanh\left(\frac{t}{m} x_k x_l\right) \tag{E.12}$$

Comme le terme $\sum_{\substack{k,l=1 \\ k\neq l}}^{N} x_k x_l \tanh\left(\frac{t}{m} x_k x_l\right)$ est toujours du signe de t, toute solution de l'équation vérifie bien la contrainte $t > 0$. De plus, $\frac{\partial^2 \ln f(t)}{\partial t^2} = \frac{1}{m} \sum_{\substack{k,l=1 \\ k\neq l}}^{N} \frac{(x_k x_l)^2}{\cosh^2\left(\frac{t}{m} x_k x_l\right)} \geq 0$, donc une solution à (E.12) sera bien un minimum de $f(t)$.

Première approximation

Résoudre cette équation n'est cependant pas aisé. On essaie donc d'en trouver la solution en faisant l'approximation la plus simple : $\tanh(x) = x$ qui est relativement bonne (moins de 10% d'erreur pour $|x| < 0.5$). On obtient alors comme minimum

$$t_0 = \frac{m\delta\|x\|^2}{\sum_{\substack{k,l=1 \\ k\neq l}}^{N} (x_k x_l)^2} \tag{E.13}$$

On note que de manière très similaire, nous avons :

$$P(\|\Phi x\|^2 \leq \|x\|^2(1-\delta)) \leq \mathbb{E}\left\{e^{t\|\Phi x\|^2 - t\|x\|^2(1-\delta)}\right\} \tag{E.14}$$

$$\leq e^{t\delta\|x\|^2} \prod_{\substack{k,l=1 \\ k\neq l}}^{N} \left(\frac{e^{\frac{t}{m}x_k x_l} + e^{-\frac{t}{m}x_k x_l}}{2}\right)^m \quad \text{pour } t < 0 \tag{E.15}$$

La recherche du t qui minimise la borne donne :

$$t_0' = \frac{-m\delta\|x\|^2}{\sum_{\substack{k,l=1 \\ k\neq l}}^{N} (x_k x_l)^2} = -t_0. \tag{E.16}$$

On obtient alors :

$$P(\|\Phi x\|^2 \geq \|x\|^2(1+\delta)) \leq e^{\frac{-m\delta^2\|x\|^4}{\sum_{\substack{k,l=1\\k\neq l}}^{N}(x_k x_l)^2}} \prod_{\substack{k,l=1\\k\neq l}}^{N} \cosh^m\left(\frac{m\delta\|x\|^2 x_k x_l}{\sum_{\substack{k,l=1\\k\neq l}}^{N}(x_k x_l)^2}\right) \tag{E.17}$$

$$P(\|\Phi x\|^2 \leq \|x\|^2(1-\delta)) \leq e^{\frac{-m\delta^2\|x\|^4}{\sum_{\substack{k,l=1\\k\neq l}}^{N}(x_k x_l)^2}} \prod_{\substack{k,l=1\\k\neq l}}^{N} \cosh^m\left(\frac{m\delta\|x\|^2 x_k x_l}{\sum_{\substack{k,l=1\\k\neq l}}^{N}(x_k x_l)^2}\right). \tag{E.18}$$

D'où le résultat suivant :

$$P\left(\left|\frac{\|\Phi x\|^2}{\|x\|^2} - 1\right| \leq \delta\right) \leq e^{\frac{-m\delta^2\|x\|^4}{\sum_{\substack{k,l=1\\k\neq l}}^{N}(x_k x_l)^2}} \prod_{\substack{k,l=1\\k\neq l}}^{N} \cosh^m\left(\frac{m\delta\|x\|^2 x_k x_l}{\sum_{\substack{k,l=1\\k\neq l}}^{N}(x_k x_l)^2}\right). \tag{E.19}$$

En pratique, la figure E.1 montre que la borne calculée par cette méthode est bien plus proche que celle du résultat proposé page 91. Cependant, lorsque le nombre de projections est élevé, on constate que la borne estimée est supérieure à la valeur mesurée empiriquement pour les δ proche de 1 (même si on s'intéresse plutôt au cas proche de 0 pour que la distorsion reste faible).

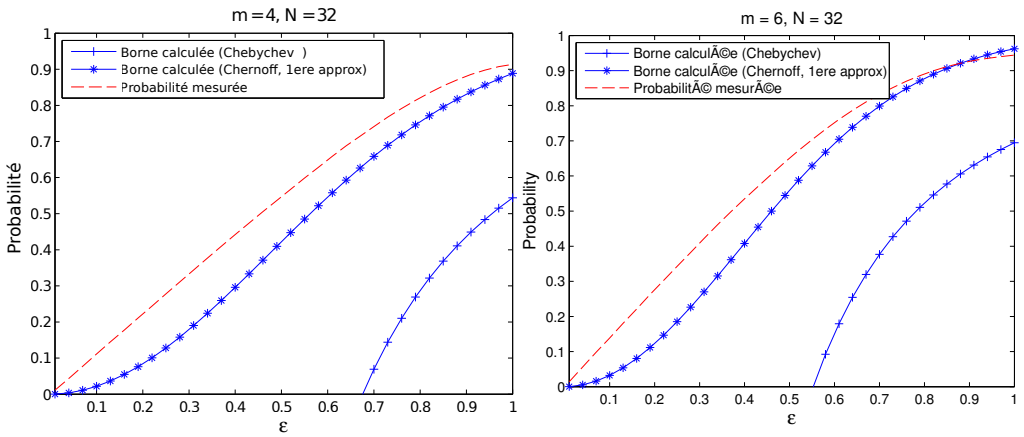

FIGURE E.1 – Borne calculée avec l'équation (7.8), celle calculée avec (E.19) et la valeur mesurée empiriquement. A gauche, $m = 4$ et à droite $m = 6$.

Meilleure approximation

Pour améliorer le résultat, on essaie une approximation plus fine de $\tanh(x)$ par $x - \frac{x^3}{3}$, l'erreur d'approximation est alors inférieure à 1% pour $x = 0.5$. L'équation (E.12) se

transforme alors en :

$$0 = -\delta\|x\|^2 + \sum_{\substack{k,l=1 \\ k\neq l}}^{N} \left(\frac{t}{m}(x_k x_l)^2 - \frac{t^3}{3m^3}(x_k x_l)^4 \right), \tag{E.20}$$

qui peut s'écrire sous la forme $t^3 + \alpha t + \beta = 0$ avec

$$\alpha = -3m^2 \frac{\sum_{\substack{k,l=1 \\ k\neq l}}^{N} (x_k x_l)^2}{\sum_{\substack{k,l=1 \\ k\neq l}}^{N} (x_k x_l)^4}; \tag{E.21}$$

$$\beta = \frac{3\delta m^3 \|x\|^2}{\sum_{\substack{k,l=1 \\ k\neq l}}^{N} (x_k x_l)^4}. \tag{E.22}$$

Cette équation peut se résoudre avec la méthode de Cardan, dont les solutions dépendent du signe de $\Delta = \beta^2 + \frac{4}{27}\alpha^3$.

Cependant, les résultats obtenus par simulation, et les solutions à l'équation données par la méthode de Cardan sont telles que l'approximation $\tanh x = x - \frac{x^3}{3}$ n'est pas bonne, puisque les solutions sont telles que $x > 1$.

Bibliographie

[AEB06] M. Aharon, M. Elad, and A. Bruckstein. K-svd : An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11) :4311–4322, 2006.

[Bar07] R.G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4) :118, 2007.

[BBL+09] J.F. Beche, S. Bonnet, T. Levi, A. Noca, G. Charvet, and R. Guillemaud. Real-time adaptive discrimination threshold estimation for embedded neural signals detection. In *Neural Engineering, 2009. NER'09. 4th International IEEE/EMBS Conference on*, pages 597–600. IEEE, 2009.

[BCI+07] S. Bourguignon, H. Carfantan, J. Idier, et al. Minimisation de criteres de moindres carrés pénalisés par la norme l1 dans le cas complexe. *ACTES 21e GRETSI*, pages 1253–1256, 2007.

[BDDW08] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3) :253–263, 2008.

[BG73] A. Björck and G.H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123) :579–594, 1973.

[Bra99] K. Brandenburg. Mp3 and aac explained. In *Proc. of the AES 17th International Conference on High Quality Audio Coding*, 1999.

[BV06] A. Bertoni and G. Valentini. Ensembles based on random projections to improve the accuracy of clustering algorithms. *Neural nets 2005*, 3931 :31, 2006.

[BW09] R.G. Baraniuk and M.B. Wakin. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 9(1) :51–77, 2009.

[BZJ10] M. Babaie-Zadeh and C. Jutten. On the stable recovery of the sparsest overcomplete representations in presence of noise. *Signal Processing, IEEE Transactions on*, 58(10) :5397–5400, 2010.

[Can08] E.J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes rendus-Mathématique*, 346(9-10) :589–592, 2008.

[CDS01] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1) :129–159, 2001.

[Cha07a] Rick Chartrand. Exact reconstructions of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14 :707–710, 2007.

[Cha07b] Rick Chartrand. Nonconvex compressed sensing and error correction. In *32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[CHDM12] B. Coppa, R. Héliot, D. David, and O. Michel. Classification from compressive representations of data. *Proceedings of EUSIPCO 2012*, 2012.

[CHJ10] R. Calderbank, S. Howard, and S. Jafarpour. Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2) :358–374, 2010.

[CHM+12] B. Coppa, R. Héliot, O. Michel, E. Moisan, and D. David. Low-cost intracortical spiking recordings compression with classification abilities for implanted bmi devices. *IEEE Engineering in Medicine and Biology Society IEEE, EMBC 2012*, 2012.

[CL12] A. Cardoso Lapolli. Hardware dédié pour la compression de données par codage parcimonieux. Rapport de stage, CEA-Léti, 2012.

[CR05] E. Candes and J. Romberg. l1-magic : Recovery of sparse signals via convex programming. *California Institute of Technology, Tech. Rep*, 2005.

[CRT06a] EJ Candes, J. Romberg, and T. Tao. Robust uncertainty principles : Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2) :489–509, 2006.

[CRT06b] E.J. Candès, J.K. Romberg, and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Communications on Pure and Applied Mathematics*, 59 :1207–1223, 2006.

[CS08] Rick Chartrand and Valentina Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24(035020) :1–14, 2008.

[CT05] EJ Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12) :4203–4215, 2005.

[CT06] EJ Candes and T. Tao. Near-optimal signal recovery from random projections : Universal encoding strategies ? *IEEE Transactions on Information Theory*, 52(12) :5406–5425, 2006.

[CY08] R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, pages 3869–3872, 2008.

[Das99] S. Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 634 –644, 1999.

[Das00] S. Dasgupta. Experiments with random projection. In *Uncertainty in Artificial Intelligence : Proceedings of the Sixteenth Conference (UAI-2000)*, pages 143–151, 2000.

[DDWB06] MF Duarte, MA Davenport, MB Wakin, and RG Baraniuk. Sparse Signal Detection from Incoherent Projections. In *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, volume 3, 2006.

[DE03] D.L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5) :2197, 2003.

[Dev] Analog Devices. Tutorial : Mixers and modulator.

[DH02] D.L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *Information Theory, IEEE Transactions on*, 47(7) :2845–2862, 2002.

[Don06] D.L. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6) :797–829, 2006.

[DT96] RA DeVore and VN Temlyakov. Some remarks on greedy algorithms. *Advances in computational Mathematics*, 5(1) :173–187, 1996.

[EB02] M. Elad and A.M. Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *Information Theory, IEEE Transactions on*, 48(9) :2558–2567, 2002.

[Ela07] M. Elad. Optimized projections for compressed sensing. *Signal Processing, IEEE Transactions on*, 55(12) :5695–5702, 2007.

[EY36] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3) :211–218, 1936.

[FB03] X.Z. Fern and C.E. Brodley. Random projection for high dimensional data clustering : A cluster ensemble approach. In *Proceedings of 20th International Conference on Machine learning*, 2003.

[GGR95] I.F. Gorodnitsky, J.S. George, and B.D. Rao. Neuromagnetic source imaging with focuss : a recursive weighted minimum norm algorithm. *Electroencephalography and clinical Neurophysiology*, 95(4) :231–251, 1995.

[Gil68] B. Gilbert. A precise four-quadrant multiplier with subnanosecond response. *Solid-State Circuits, IEEE Journal of*, 3(4) :365–373, 1968.

[GMCH11] L. Galluccio, O.J.J. Michel, P. Comon, and A.O. Hero. Graph based k-means clustering. *Elsevier Signal Processing*, dec 2011.

[GN03] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *Information Theory, IEEE Transactions on*, 49(12) :3320–3325, 2003.

[GNL+08] F Galan, M Nuttin, E Lew, P W Ferrez, G Vanacker, J Philips, and J R Millan. A brain-actuated wheelchair : Asynchronous and non-invasive Brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119 :2159–2169, 2008.

[Gol68] R. Gold. Maximal recursive sequences with 3-valued recursive cross-correlation functions (corresp.). *Information Theory, IEEE Transactions on*, 14(1) :154–156, 1968.

[GR97] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using FOCUSS : A re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on*, 45(3) :600–616, 1997.

[HSF+06] Leigh R Hochberg, Mijail D Serruya, Gerhard M Friehs, Jon A Mukand, Maryam Saleh, Abraham H Caplan, Almut Branner, David Chen, Richard D Penn, and John P Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(July) :164–171, 2006.

[JL84] W.B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206) :1–1, 1984.

[Kas66] T. Kasami. Weight distribution formula for some class of cyclic codes, 1966.

[KDMR+03] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2) :349–396, 2003.

[KLW+07] S. Kirolos, J. Laska, M. Wakin, M. Duarte, D. Baron, T. Ragheb, Y. Massoud, and R. Baraniuk. Analog-to-information conversion via random demodulation. In *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, pages 71–74. IEEE, 2007.

[Kou10] B. Kouassi. Etude d'un convertisseur analogique information (aic) par un modulateur aléatoire. Rapport de master, CEA-Léti, 2010.

[Lew98] MS Lewicki. A review of methods for spike sorting : the detection and classification of neural action potentials. *Network : Computation in Neural Systems*, 9(4) :R53–R78, 1998.

[LKD+07] J.N. Laska, S. Kirolos, M.F. Duarte, T.S. Ragheb, R.G. Baraniuk, and Y. Massoud. Theory and implementation of an analog-to-information converter using random demodulation. In *Proceedings of the IEEE International Symposium on Circuites and Systems (ISCAS), New Orleans, Louisiana*. Citeseer, 2007.

[Llo82] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2) :129 – 137, mar 1982.

[Mal08] S. Mallat. A wavelet tour of signal processing, third edition : The sparse way, 2008.

[MBP+08] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *arXiv preprint arXiv :0809.3083*, 2008.

[MBZJ09] H. Mohimani, M. Babaie-Zadeh, and C. Jutten. A fast approach for overcomplete sparse decomposition based on smoothed< formula formulatype=. *Signal Processing, IEEE Transactions on*, 57(1) :289–301, 2009.

[ME09a] M. Mishali and Y.C. Eldar. Expected rip : Conditioning of the modulated wideband converter. In *Information Theory Workshop, 2009. ITW 2009. IEEE*, pages 343–347. IEEE, 2009.

[ME09b] M. Mishali and Y.C. Eldar. Expected RIP : Conditioning of The Modulated Wideband Converter. *CCIT Report no. 736*, 2009.

[ME10] M. Mishali and Y.C. Eldar. From Theory to Practice : Sub-Nyquist Sampling of Sparse Wideband Analog Signals. *IEEE Journal of Selected Topics in Signal Procesing*, 4(2) :375, 2010.

[MEDS09] M. Mishali, Y.C. Eldar, O. Dounaevsky, and E. Shoshan. Xampling : Analog to digital at sub-nyquist rates. *Arxiv preprint arXiv :0912.2495*, 2009.

[MZ93] SG Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12) :3397–3415, 1993.

[Nat95] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2) :227–234, 1995.

[Nee09] D. Needell. *Topics in Compressed Sensing*. PhD thesis, University of California, 2009.

[O+96] B.A. Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583) :607–609, 1996.

[OF97] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set : A strategy employed by v1 ? *Vision research*, 37(23) :3311–3325, 1997.

[PL11] S. Park and H.N. Lee. On the derivation of rip for random gaussian matrices and binary sparse signals. In *ICT Convergence (ICTC), 2011 International Conference on*, pages 120–124. IEEE, 2011.

[PRK93] Y.C. Pati, R. Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[QQNBS04] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, 16(8) :1661–1667, 2004.

[RKL+08] T. Ragheb, S. Kirolos, J. Laska, A. Gilbert, M. Strauss, R. Baraniuk, and Y. Massoud. Implementation models for analog-to-information conversion

via random sampling. In *Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on*, pages 325–328. IEEE, 2008.

[SCE01] A. Skodras, C. Christopoulos, and T. Ebrahimi. The jpeg 2000 still image compression standard. *Signal Processing Magazine, IEEE*, 18(5) :36–58, 2001.

[Shl94] S. Shlien. Guide to mpeg-1 audio standard. *Broadcasting, IEEE Transactions on*, 40(4) :206–218, 1994.

[SMA+08] G Schalk, K J Miller, N R Anderson, J A Wilson, M D Smyth, J G Ojemann, D W Moran, J R Wolpaw, and E C Leuthardt. Two-dimensional movement control using electrocorticographic signals in humans. *Journal of neural Engineering*, 5 :75–84, 2008.

[TG07] J.A. Tropp and A.C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12) :4655, 2007.

[Tik63] A. Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, volume 5, page 1035, 1963.

[TLD+09] J.A. Tropp, J.N. Laska, M.F. Duarte, J.K. Romberg, and R.G. Baraniuk. Beyond nyquist : Efficient sampling of sparse bandlimited signals. *Information Theory, IEEE Transactions on*, 56(1) :520–544, 2009.

[TW04] M.D. Tsai and H. Wang. A 0.3-25-GHz ultra-wideband mixer using commercial 0.18-μm CMOS technology. *Microwave and Wireless Components Letters, IEEE*, 14(11) :522–524, 2004.

[VPS+08] Meel Velliste, Sagi Perel, M Chance Spalding, Andrew S Whitford, and Andrew B Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453(June) :1098–1101, 2008.

[Wal91] G.K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4) :30–44, 1991.

Résumé

Le codage parcimonieux permet la reconstruction d'un signal à partir de quelques projections linéaires de celui-ci, sous l'hypothèse que le signal se décompose de manière parcimonieuse, c'est-à-dire avec peu de coefficients, sur un dictionnaire connu. Le codage est simple, et la complexité est déportée sur la reconstruction. Après une explication détaillée du fonctionnement du codage parcimonieux, une présentation de quelques résultats théoriques et quelques simulations pour cerner les performances envisageables, nous nous intéressons à trois problèmes : d'abord, l'étude de conception d'un système permettant le codage d'un signal par une matrice binaire, et des avantages apportés par une telle implémentation. Ensuite, nous nous intéressons à la détermination du dictionnaire de représentation parcimonieuse du signal par des méthodes d'apprentissage. Enfin, nous discutons la possibilité d'effectuer des opérations comme la classification sur le signal sans le reconstruire.

Mots-clés : codage parcimonieux, minimisation ℓ_1, IRLS, Matching Pursuit, classification, compression, apprentissage de dictionnaire

---

Abstract

Compressed sensing allows to reconstruct a signal from a few linear projections, under the assumption that the signal can be sparsely represented, that is, with only a few coefficients, on a known dictionary. Coding is very simple and all the complexity is gathered on the reconstruction. After more detailed explanations of the principle of compressed sensing, some theoretic resultats from literature and a few simulations allowing to get an idea of expected performances, we focuss on three problems : First, the study for the building of a system using compressed sensing with a binary matrix and the obtained benefits. Then, we have a look at the building of a dictionary for sparse representations of the signal. And lastly, we discuss the possibility of processing signal without reconstruction, with an example in classification.

Keywords : compressed sensing, ℓ_1-minimization, IRLS, Matching Pursuit, classification, compression, dictionary learning

---