

# Table des matières

<b>1</b>	<b>Contexte, problématiques et hypothèses du manuscrit</b>	<b>5</b>
1.1	Historique des télécommunications . . . . .	5
1.1.1	Systèmes optiques avec intervention humaine . . . . .	6
1.1.2	Télécommunications câblées électriques . . . . .	6
1.1.3	Essors des télécommunications sans-fil . . . . .	6
1.1.4	Téléphonie mobile moderne . . . . .	7
1.1.5	Une nouvelle dimension de multiplexage : l'espace . . . . .	7
1.2	Décomposition d'une liaison MIMO . . . . .	8
1.2.1	Hypothèses générales et cas d'utilisation pratiques . . . . .	9
1.2.2	Canal de transmission . . . . .	11
1.2.3	Modulation et démodulation . . . . .	14
1.2.4	Codage et décodage . . . . .	14
1.2.5	Synthèse de la modélisation mathématique . . . . .	16
1.3	État de l'art des détecteurs MIMO . . . . .	17
1.3.1	Détecteur par maximum de vraisemblance . . . . .	19
1.3.2	Détecteurs linéaires . . . . .	19
1.3.3	Détecteurs à annulation des interférences . . . . .	20
1.3.4	Détecteurs basés sur un parcours d'arborescence . . . . .	24
1.3.5	Apprentissage par réseaux de neurones . . . . .	32
1.3.6	Détecteurs bio-inspirés . . . . .	32
1.3.7	Détecteurs géométriques . . . . .	34
1.4	Synthèse de l'état de l'art et problématique des travaux . . . . .	35
1.4.1	Synthèse des stratégies de détection . . . . .	35
1.4.2	Différence entre le problème de détection théorique et celui résolu en pratique . . . . .	37
1.4.3	Problématique et structure du manuscrit : de l'intérêt de l'exploration pour la détection MIMO . . . . .	38
<b>2</b>	<b>Étude d'un détecteur à parcours d'arborescence stochastique</b>	<b>39</b>
2.1	Méthodes et critères d'évaluation . . . . .	39
2.1.1	BER en fonction du SNR : la méthode de Monte Carlo . . . . .	40
2.1.2	Évaluation de la complexité arithmétique . . . . .	41
2.1.3	Analyse de Pareto . . . . .	42
2.2	Vocabulaire du parcours d'arborescence . . . . .	43
2.3	Un nouveau détecteur à parcours d'arborescence stochastique . . . . .	45
2.3.1	De la métaheuristique bio-inspirée au parcours d'arborescence stochastique . . . . .	45
2.3.2	Proposition d'un nouveau détecteur basé sur le FA . . . . .	47

## Table des matières

2.3.3	Synthèse du détecteur par parcours d'arborescence stochastique .	50
2.4	Les références à parcours d'arborescence . . . . .	52
2.4.1	Le parcours en largeur canonique : le K-best de [3] . . . . .	52
2.4.2	Le parcours parallèle par métrique de [4] . . . . .	53
2.5	Comparaison du parcours d'arborescence stochastique avec des références	55
2.5.1	Surcoût de complexité engendré par l'exploration . . . . .	55
2.5.2	Comparaison des performances au sens de Pareto . . . . .	58
2.6	Conclusion sur la détection par parcours d'arborescence stochastique .	61
<b>3</b>	<b>Diagnostic du détecteur géométrique historique</b>	<b>63</b>
3.1	Préambule mathématique . . . . .	63
3.1.1	Expression du modèle de transmission dans le domaine réel . . .	64
3.1.2	Prétraitement par décomposition en valeurs singulières . . . . .	64
3.1.3	Obtention de la pseudo-inverse de Moore-Penrose . . . . .	67
3.2	La détection géométrique historique . . . . .	67
3.2.1	Recherche générale : exploration géométrique . . . . .	68
3.2.2	Recherche locale : exploitation . . . . .	70
3.2.3	Positionnement du détecteur géométrique dans la taxonomie ex- ploration/exploitation . . . . .	71
3.3	Atouts et faiblesses de la version historique . . . . .	71
3.3.1	Analyse de la complexité arithmétique . . . . .	71
3.3.2	Performances de détection . . . . .	72
3.3.3	Implémentabilité sur architecture matérielle . . . . .	74
3.4	Synthèse sur la détection géométrique historique : une stratégie promet- teuse et des défauts à améliorer . . . . .	78
<b>4</b>	<b>Améliorations, études et implémentation de la détection géométrique</b>	<b>79</b>
4.1	Améliorations algorithmiques de la détection géométrique . . . . .	79
4.1.1	Majoration du nombre d'itérations . . . . .	80
4.1.2	Projection sur les sous-constellations et sorties souples . . . . .	81
4.1.3	Retirer les intersections pour réduire la complexité . . . . .	83
4.1.4	Opérations au niveau bloc et au niveau vecteur . . . . .	85
4.1.5	Réutilisation des fonctions objectifs . . . . .	87
4.2	Analyses de Pareto des améliorations . . . . .	89
4.2.1	Complexité arithmétique asymptotique . . . . .	89
4.2.2	Analyses de Pareto . . . . .	90
4.3	Implémentation matérielle par synthèse haut-niveau . . . . .	93
4.3.1	Sélection des paramètres et des cas d'utilisation . . . . .	95
4.3.2	Évaluation optimisée de la fonction objectif des points de $\mathcal{P}$ . . .	95
4.3.3	Quantification des variables . . . . .	97
4.3.4	Présentation de la synthèse haut-niveau . . . . .	98
4.3.5	Architecture de l'implémentation . . . . .	99
4.3.6	Résultats de l'implémentation . . . . .	101
4.4	Conclusion sur les améliorations du détecteur géométrique . . . . .	104

<b>5 Synthèse des travaux et perspectives</b>	<b>105</b>
5.1 Synthèse des travaux et contributions . . . . .	105
5.2 Conclusion . . . . .	106
5.3 Perspectives . . . . .	107





# Remerciements

Le long de ces deux pages, je voudrais remercier toutes les personnes qui m'ont offert du temps, du soutien et de l'énergie pour faire aboutir ce projet de thèse. Même si j'en suis évidemment l'auteur principal, ce manuscrit reste à bien des égards une œuvre collective qui n'aurait pas existé sous cette forme sans l'apport nombreux intervenants.

Avant de remercier les acteurs de cette thèse, je souhaite citer ici ceux qui m'ont poussé à croire en moi et à m'engager dans la carrière scientifique. Ainsi, Patrick Calaïni a eu un rôle fondamental dans mon épanouissement et dans mes choix de vies. De même, M. Escaffre ainsi que mes autres enseignants ont profondément impacté mes choix de carrière. De plus, je souhaite noter, bien que ce soit atypique, les grands penseurs qui m'ont aidé à construire ma conception de la science et de la méthode scientifique : John Dewey, William James, Bruno Latour, Karl Popper et Gaston Bachelard notamment. Enfin, je remercie mes formateurs à l'ENS Rennes qui m'ont permis de me développer en tant que scientifique et chercheur : Martinus Werts, Bernard Multon, Roman Legoff-Latimier et tous les autres membres du département mécatronique.

Je remercie évidemment l'équipe SCEE qui m'a hébergé pendant mon stage de Master ainsi que pendant toute la préparation de ma thèse. Je remercie mon encadrant, Amor Nafkha, qui a toujours cru en moi au point de me confier la poursuite de ses travaux de thèse et de me laisser explorer les pistes de recherches à ma guise. Je remercie aussi mon directeur de thèse, Yves Louët, pour avoir toujours répondu présent, et ce d'autant plus pendant la difficile période de rédaction. Son intérêt pour l'histoire des sciences et sa culture des télécommunications ont permis des discussions passionnantes et ont profondément influencé le premier chapitre de ce manuscrit. Je remercie aussi les autres permanents de l'équipe avec qui nous avons pu échanger sur divers sujets : implémentation matérielle avec Rubén Salvador, enseignement des codes correcteurs avec Haïfa Fares...

Je remercie ensuite toutes les personnes avec qui j'ai échangé lors des pauses cafés, des repas et en bien d'autres occasions : Rémi Bonnefoi, Lilian Besson, Morgane Casanova, Corentin Guezenoc, Simon Leglaive, Karim Kassan, Adrien Llave, Nabil Zaraneh... En plus d'être très agréables, toutes ces discussions m'ont introduit au monde de la recherche, à l'institution de CentraleSupélec et m'ont fait avancer sur de nombreux sujets. Un merci particulier à Adrien avec qui j'ai eu la chance de travailler quotidiennement. Sa maïeutique a permis de raffiner ma compréhension de la réception MIMO bien au-delà de ce que j'aurais pu faire seul. Nos débats en tout genre m'ont également apporté beaucoup. De plus, il a apporté son expertise technique pour l'enregistrement et la diffusion de ma soutenance.

En plus des chercheurs du campus de Rennes, je remercie également le reste du personnel : Karine Bernard pour son efficacité et sa gentillesse, Maud Julien pour les

## *Table des matières*

parenthèses au potager partagé ainsi que le personnel de la restauration qui a toujours offert un accueil chaleureux.

La préparation de ma thèse m'a également permis d'enseigner à l'ENS Rennes. Je remercie tous mes collègues du département mécatronique, ainsi que mes élèves. Ces temps d'enseignement m'ont permis de changer d'air tout en mettant en lumière les synergies entre recherche et enseignement. Je remercie particulièrement Youness Akourim, Arthur Poquet, Adrien Gaggioli et Valentin Burgaud qui ont effectué des stages ou des projets de recherche avec moi. Leurs études ont significativement aidé à forger les résultats de cette thèse.

Parce qu'une thèse requiert au moins autant des soutiens moraux que des apports techniques, je me dois de remercier tous ceux qui m'ont apporté leur aide et avec qui j'ai pu partager les moments joyeux comme difficiles. Tout d'abord, mes anciens collègues de promo qui ont traversé les mêmes difficultés et avec qui nous avons pu échanger des astuces : Baptiste Rebouillat, Léna Lejeune, Marie-Charlotte Rivière, Guillaume Nordet, Sylvain Charlès et les autres. Ensuite ma famille qui m'a soutenu et supporté jusqu'au bout. En particulier, je remercie ma mère Christine Gibelin pour les relectures intégrales du manuscrit qui ont permis de détecter énormément de typos. Je remercie aussi mon frère Clément Trotobas pour ses conseils qui ont souvent apporté un nouveau regard sur les problématiques rencontrées.

Je remercie aussi tous ceux qui m'ont proposé des temps d'évasion pour oublier un temps les labeurs de la recherche. Ainsi, l'équipe de volley du BDS et son entraîneur Olivier Kirzin m'ont offert un exutoire hebdomadaire et un réel sentiment de camaraderie. De même, mes deux tables deux jeux de rôles (Clément/Faret, Vincent/Kyogen, Arnaud/Kerak, Loric/Tasos, Jehan/Jalfer et Adrien/Lorem) m'ont permis de m'évader vers des aventures héroïques et de grands éclats de rire. Je remercie du fond du cœur la famille Bourreau-Charlès (Sylvain, Louise, Alice, Lilou, Noé et mon filleul Robin) de m'avoir accueilli aussi chaleureusement dans leur famille. Les journées passées à leur côté et leur soutien indéfectible n'ont pas de prix.

Je terminerai les remerciements par ceux qui ont été à la conclusion de ce périple : le jury de ma soutenance. Les retours et les discussions ont été très intéressantes et stimulantes tout en étant particulièrement bienveillantes et amicales. Ça a été un plaisir de soutenir devant un tel jury.

Enfin, je remercie dans un dernier paragraphe ma compagne Claire Livet que je ne peux classer dans aucune des catégories précédentes tant elle m'a apporté. En effet, elle pourrait être citée dans chacune d'entre elle. Ainsi, elle m'a beaucoup apporté sur les plans scientifiques et techniques comme sur la relecture du manuscrit. Elle m'a aussi apporté confiance, stabilité, soutien et divertissement durant les trois années de préparation. Enfin, elle a significativement amélioré ma soutenance de thèse par ses conseils avisés.

# Introduction

L'histoire des télécommunications est marquée par l'augmentation coordonnée des performances (débit de transmission, robustesse, portée, efficacité énergétique...) et de la complexité des systèmes utilisés. L'essor des systèmes à entrées et sorties multiples (multiple-input multiple-output, MIMO) ne déroge pas à cette règle. En effet, leur inclusion dans de nombreux standards modernes (WiFi, WiMAX, 3G, 4G, 5G) a permis des améliorations de performances considérables mais a également complexifié les algorithmes de réception. Ainsi, ces systèmes requièrent des détecteurs pour compenser les évanouissements et les bruits induits par le canal de transmission ainsi que pour séparer les signaux provenant des différentes antennes émettrices.

Une grande diversité existe dans les stratégies de détection proposées par la littérature. La majorité d'entre elles exploitent les connaissances que le récepteur possède à travers des modèles et l'estimation de l'état du canal. Cependant, ces approches peuvent être mises en défaut notamment lorsque le canal s'éloigne du modèle adopté, lorsqu'il est trop bruité ou mal estimé. Les métaheuristiques bio-inspirées sont connues pour éviter ces écueils en explorant davantage l'espace des possibles mais cela implique également un surcoût de complexité important.

Cette thèse examine le compromis entre les deux approches : l'exploitation des modèles théoriques et l'exploration. Dans un premier temps, l'importance de l'exploration est mise en lumière et nous examinons son impact sur les performances et sur la complexité du détecteur. Par la suite, deux nouveaux détecteurs mixant exploration et exploitation sont développés et comparés à des détecteurs de référence de la littérature n'utilisant que de l'exploitation. Enfin ces détecteurs, et à travers eux le compromis exploration-exploitation, sont étudiés en termes de performance et de complexité depuis l'élaboration algorithmique jusqu'à l'implémentation sur réseau de portes programmables (field-programmable gate array, FPGA).

## Structure du manuscrit

Le chapitre 1 présente le contexte des travaux présentés dans ce manuscrit. Ce contexte comprend d'une part un historique des télécommunications et d'autre part une décomposition de la chaîne de transmission. Ce panorama permet de lister les objectifs à poursuivre ainsi que les facteurs influençant les détecteurs que nous étudierons. De plus, un état de l'art des algorithmes de détection est proposé sur la base de celui que nous avons publié dans [1]. Cette revue classe les détecteurs en fonction de la stratégie qu'ils utilisent, ce qui complète le contexte précédent. Le chapitre se conclut en dégagant de tous les éléments une hypothèse de travail ainsi qu'une méthode de vérification qui sera mise en œuvre dans les chapitres suivants. L'hypothèse du manuscrit est alors que l'exploration rend les détecteurs MIMO plus résilients.

## Table des matières

Le chapitre 2 évalue la véracité de l'hypothèse en comparant un nouveau détecteur avec des références proches. Pour ce faire, nous définissons des critères de comparaison et une méthode d'analyse objective. En particulier, l'analyse de Pareto est présentée pour étudier le compromis performance/complexité. De plus, un nouveau détecteur à parcours d'arborescence est développé pour rester proche de la littérature tout en ajoutant une composante d'exploration. Ainsi, le détecteur bio-inspiré basé sur l'algorithme des lucioles (firefly algorithm, FA) de [2] est présenté puis traduit dans le paradigme de parcours d'arborescence. Notre nouveau détecteur généralise également l'algorithme présenté dans [2] pour produire des sorties souples et accepter toutes les modulations. Enfin, une analyse de Pareto entre notre nouveau détecteur et deux références par parcours d'arborescence [3, 4] permet de conclure sur la pertinence d'ajouter de l'exploration. L'ensemble de ce chapitre s'appuie sur les travaux que nous avons publiés dans [5].

Une fois l'intérêt de l'exploration mis en lumière, le chapitre 3 introduit un détecteur entièrement conçu autour du compromis exploration-exploitation : le détecteur géométrique [6, 7]. Puisque cette stratégie intègre l'exploration dès sa conception, plutôt que de l'ajouter tardivement comme dans le cas de notre détecteur basé sur le FA, nous espérons observer plus de bénéfices à explorer. Ce chapitre propose donc un diagnostic complet de la stratégie géométrique en se basant sur nos travaux publiés dans [8]. Ainsi, nous introduisons une formalisation mathématique, une analyse de la complexité arithmétique et une évaluation des performances de détection. De plus, un prototype implémenté sur un système sur puce (system on chip, SoC) permet d'identifier les atouts et les défauts de la stratégie en terme d'implémentabilité matérielle.

Le chapitre 4 commence par présenter les améliorations de la détection historique que nous avons publié dans [9]. L'objectif est de lever les défauts mis en lumière au chapitre 3 tout en accentuant les avantages. Plusieurs analyses comparatives avec des détecteurs de la littérature permettent ensuite de valider les résultats de ces améliorations. Enfin, le nouveau détecteur est implémenté sur FPGA et des comparaisons en termes d'utilisation de ressource, de débit et de consommation énergétique sont réalisées.

Enfin, le chapitre 5 synthétise les études et les développements présentés dans le manuscrit. Ceux-ci sont remis en perspective pour dégager des résultats généraux et pour proposer des perspectives de recherches.

## Notations mathématiques

Dans ce manuscrit, les majuscules (*resp.* minuscules) en gras représentent les matrices (*resp.* les vecteurs). Les ensembles sont notés par une lettre calligraphique capitale et la lettre minuscule correspondante précise le cardinal de cet ensemble. Le tableau ci-dessous liste les autres notations et leur définition.

Notation	Définition
$\Re(a)$	Partie réelle du complexe $a$
$\Im(a)$	Partie imaginaire du complexe $a$
$\ \mathbf{y}\ $	Norme euclidienne du vecteur $\mathbf{y}$
$\ \mathbf{H}\ _F$	Norme de Frobenius de $\mathbf{H}$
$\mathbf{H}^T$	Transposée de $\mathbf{H}$
$\mathbf{H}^H$	Transposée hermitienne de $\mathbf{H}$
$\mathbf{H}^+$	Pseudo-inverse de Moore-Penrose
$\mathbf{h}_j$	$j^{\text{ème}}$ colonne de $\mathbf{H}$
$y_i$	$i^{\text{ème}}$ coefficient du vecteur $\mathbf{y}$
$\mathbb{E}$	Espérance d'une variable aléatoire
$\llbracket a, b \rrbracket$	Ensemble des entiers compris entre $a \in \mathbb{N}$ et $b \in \mathbb{N}$ inclus
$\text{diag}(\mathbf{y})$	Matrice diagonale de diagonale le vecteur $\mathbf{y}$
$\text{card}(\mathcal{P})$	Cardinal de l'ensemble $\mathcal{P}$
$\mathbb{C}$	Ensemble des complexes
$\mathbb{R}$	Ensemble des réels
$\mathbb{R}_+$	Ensemble des réels positifs
$\mathbb{N}$	Ensemble des entiers naturels
$\mathbb{Z}$	Ensemble des entiers relatifs
$\bar{R}$	Négation de la variable $R$ en algèbre de Boole
$\oplus$	Ou exclusif de l'algèbre de Boole



# 1 Contexte, problématiques et hypothèses du manuscrit

La théorie de l'enquête scientifique affirme que la première étape d'un travail scientifique doit toujours être d'analyser les théories existantes pour identifier les besoins d'amélioration, de formuler une ou plusieurs hypothèses et de définir une ou plusieurs méthodes permettant de tester la validité de ces hypothèses. Ce chapitre traite donc de ces trois étapes fondamentales.

Pour ce faire, la section 1.1 présente un rapide historique des télécommunications depuis les signaux de fumée pré-colombiens jusqu'aux systèmes mobiles modernes. Cette histoire permet de mettre en lumière les objectifs qui motivent le développement des nouvelles technologies de télécommunications.

Une fois l'objectif fixé, la section 1.2 décrit les composants élémentaires de la chaîne de transmission MIMO. Cette partie présente aussi l'interaction entre composants qui assemblent un système transmettant une information à travers un canal MIMO. Cela permet de lister tous les facteurs influents sur les algorithmes de détection qui sont l'objet de ces travaux de thèse.

La section 1.3 liste les différentes solutions mises en œuvre dans la littérature pour résoudre le problème de détection MIMO. Cet état de l'art permet de formaliser les stratégies utilisées par chacun des algorithmes afin de clarifier la situation actuelle.

Enfin, la section 1.4 dégage une problématique à partir des informations précédentes. De plus, elle énonce également une hypothèse permettant de résoudre la problématique. Enfin, une méthode permettant de tester la validité de l'hypothèse est proposée.

## 1.1 Historique des télécommunications

Dans son livre fondateur, Claude Shannon définit “le problème fondamental de la communication [par le fait] de reproduire exactement ou approximativement en un point, un message choisi en un autre point” [10]. En ce sens, la communication balaie un spectre très large, depuis la parole et le langage des signes jusqu'aux pages de l'internet moderne pouvant être affichées quasiment partout. Lorsque les points d'émission et de réception sont suffisamment éloignés, on parle alors de *télécommunications* et on impose classiquement une contrainte sur la latence. Cette section dresse un historique des systèmes de télécommunication afin de mettre en lumière les principaux objectifs qui ont guidé leur développement : augmentation de la portée, du débit et de la fiabilité, réduction de la latence, des coûts financiers et énergétiques.

### 1.1.1 Systèmes optiques avec intervention humaine

Jusqu’au XVIII<sup>ème</sup> siècle, les télécommunications se basent sur la transmission de signaux optiques simples tels qu’un feu depuis un point surélevé ou des colonnes de fumées. Les signaux sont parfois répétés spatialement par l’observateur afin d’augmenter la portée générale du système. Toutes les variantes de ces systèmes se révèlent extrêmement coûteuses, surtout lorsque l’observateur doit relayer l’information. En effet, une personne attentive ainsi qu’un système de production de signal sont requis pour chaque dizaine de kilomètres à parcourir. De même, la latence est extrêmement grande en raison des interventions humaines à chaque répétition [11, chap. 8]. Enfin, ces systèmes transmettent souvent des informations très simples : une torche allumée permet de transmettre l’équivalent d’un bit d’information moderne comme la présence d’un ennemi. Les versions les plus complexes, tels que les signaux de fumée des peuples pré-colombiens, peuvent transmettre des messages un peu plus élaborés en produisant une série de signaux tout-ou-rien (présence ou absence de fumée) formant des mots dans un dictionnaire simple (“attention”, “tout va bien”, “au secours”, etc). Dans tous les cas, les débits d’information de ces systèmes ne sont jamais élevés. Les systèmes modernes pourraient par exemple transmettre ces informations avec un débit binaire de quelques bits par minute. A la fin du XVIII<sup>ème</sup> siècle et au début du siècle suivant, les télégraphes de Chappe réduisent la latence en permettant de transmettre un message à plusieurs centaines de kilomètres en quelques heures mais le débit ne dépasse toujours pas le bit par seconde [11, 12].

### 1.1.2 Télécommunications câblées électriques

Le développement de l’électricité au XIX<sup>ème</sup> siècle permet l’essor du télégraphe électrique qui transmet des impulsions électriques dans un fil en cuivre. Ce dispositif, associé au code Morse, transmet pour la première fois une dizaine de bits modernes par seconde avec deux symboles par seconde sur un alphabet de 36 symboles (26 lettres et 10 chiffres). Moins d’un siècle plus tard, le téléphone analogique surpasse son prédécesseur en transmettant directement la voix. Outre le recours au câble coaxial qui améliore le canal de transmission, c’est bien le passage au signal analogique et l’introduction du multiplexage fréquentiel qui permettent cette augmentation de débit. Jusqu’alors, un canal de transmission était requis pour chaque message et le seul multiplexage possible était temporel. L’ajout d’un second domaine de multiplexage, la fréquence, permet aux câbles coaxiaux de servir jusqu’à 300 communications audios simultanées [13].

### 1.1.3 Essors des télécommunications sans-fil

La modulation analogique et le multiplexage fréquentiel dans les câbles coaxiaux inspire également l’apparition des télécommunications sans-fil durant le XX<sup>ème</sup> siècle. D’abord dédié à la diffusion des nouvelles et des programmes ludiques par la radio puis la télévision, les guerres mondiales accélèrent l’apparition d’appareils de plus en plus portatifs d’abord restreints aux usages militaires. La miniaturisation se poursuit du-



rant l'après-guerre avec l'apparition des transistors modernes [14]. A cette période, les télécommunications sans-fil restent majoritairement analogiques malgré des avancées fondatrices comme la théorie de l'information de Claude Shannon [10], celle des codes correcteurs de Hamming [15] ou encore le fameux théorème de l'échantillonnage de Nyquist-Shannon [16, 17]. Il faudra attendre que les circuits imprimés, et avec eux les ordinateurs, prennent leur essor pour que le numérique devienne la norme dans les télécommunications sans-fil grand public.

### 1.1.4 Téléphonie mobile moderne

La téléphonie mobile utilise un réseau constitué de deux types d'intervenants : une station de base qui administre la cellule et un certain nombre de mobiles utilisateurs. La transmission est hertzienne entre les utilisateurs et la station de base puis elle devient filaire entre les stations servant la liaison. Les performances des réseaux mobiles sont donc liées à la fois à la capacité de liaison mais aussi à celle du réseau supportant les stations de base. L'essor des fibres optiques et les commutations de paquets sont des exemples de technologie ayant aidé au développement du réseau mobile bien qu'elles ne soient pas directement apparentes aux utilisateurs.

Bien que les technologies les plus anciennes ne permettent pas de passer d'une cellule à l'autre au cours d'un appel, ce type de changement à la volée fait son apparition dès les années 80 permettant le déploiement des premiers réseaux mobiles au sens moderne. Par la suite, les nouvelles générations de standards mobiles s'enchaînent toutes les dizaines d'années augmentant significativement le débit et ajoutant des fonctionnalités nouvelles. Le SMS fait par exemple son apparition dans les années 90 lors de la deuxième génération (2G), les transferts multimédia et internet sont ajoutés vers 2007 avec la troisième génération (3G). Le standard long-term evolution (LTE) de la quatrième génération (4G) augmente encore les débits montants et descendants et améliore l'efficacité spectrale grâce à de nouvelles formes d'ondes, à de nouveaux codages de source et de canal ainsi qu'à des nouvelles techniques de multiplexage fréquentiel et spatial.

### 1.1.5 Une nouvelle dimension de multiplexage : l'espace

Longtemps vu comme un frein à des communications de qualité, la propagation multi-trajet devient un atout avec la formalisation des systèmes à entrées et sorties multiples (multiple-input multiple-output, MIMO). Les premiers travaux théoriques de ce principe remontent aux années 70 mais c'est dans les années 90 que les premiers papiers et brevets décrivent des applications pratiques [18–21]. La fin des années 90 voit l'apparition des premiers prototypes fonctionnels de systèmes MIMO [22]. Le principe est assez simple : utiliser plusieurs antennes à l'émission et à la réception pour transmettre dans le même slot temps-fréquence et discriminer les flux de données à partir de considérations spatiales.

L'espace devient alors une nouvelle dimension de multiplexage permettant une amélioration considérable de l'efficacité spectrale, du débit et de la qualité des liaisons. Les codages spatio-temporels permettent par exemple de transmettre la même quantité

d'information avec les mêmes ressources fréquentielles et temporelles de façon beaucoup plus résiliente. Le code d'Alamouti spécifiquement construit pour les systèmes 2x2 [23] a ouvert la voie à des extensions aux systèmes plus larges [24, 25]. Ces codes spatio-temporels orthogonaux sont à la base de transmissions MIMO très fiables. De plus, le multiplexage spatial peut aussi être utilisé pour augmenter le débit de transmission de données en transmettant simultanément plusieurs flux de données. Enfin, un mélange des deux régimes précédents peut permettre d'augmenter à la fois la fiabilité et le débit [26]. Les systèmes MIMO démontrent donc une bonne versatilité permettant de choisir un compromis entre fiabilité et débit adapté à l'application.

De nombreuses variantes du principe initial ont été proposées dans le sillage des premiers systèmes tel que le MIMO massif dans lequel le nombre d'antennes dépasse la centaine, le MIMO à plusieurs utilisateurs ou encore le MIMO coopératif qui exploite plusieurs stations de base simultanément. Cette technologie a été intégrée dans les standards mobiles dès la 3G ainsi que dans les standards WiMAX et Wi-Fi [27, 28].

## 1.2 Décomposition d'une liaison MIMO

Plusieurs modélisations de la chaîne de transmission MIMO sont possibles selon le niveau de détail recherché et selon le point d'intérêt. Dans ce manuscrit, nous adoptons la décomposition présentée dans la figure 1.1. La suite de cette section détaille le contenu de chacun des composants présents dans ce schéma à l'exception de la détection qui fera l'objet d'une revue de littérature étendue dans la section 1.3. Cette partie se conclut par une synthèse des modélisations mathématiques en section 1.2.5.

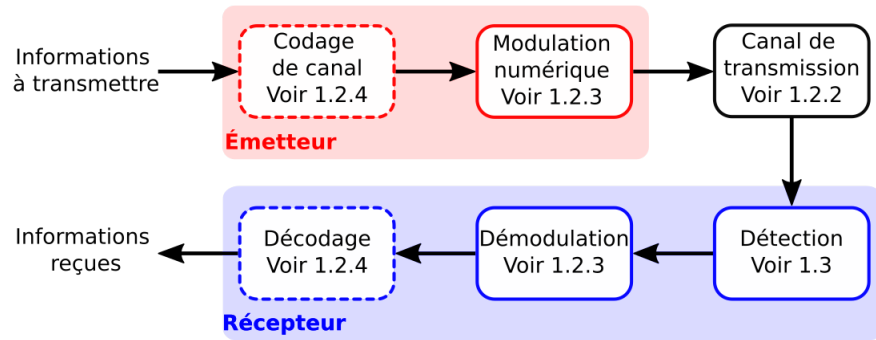


Figure 1.1) Structure schématique d'une chaîne de transmission avec en pointillé les étapes optionnelles.

### 1.2.1 Hypothèses générales et cas d'utilisation pratiques

Cette section esquisse un cadre d'application général pour contextualiser notre étude et guider les hypothèses à venir. Deux exemples de standards rentrant dans ce cadre général sont aussi introduits pour justifier de la pertinence de ces hypothèses théoriques.

#### Cadre d'application général

Ce manuscrit étudie la transmission d'informations, éventuellement codées, sur un canal MIMO hertzien. Nous supposons que le récepteur dispose d'un jeu d'antennes permettant de capter le signal à une fréquence donnée en différents points de l'espace. Aucune hypothèse n'est faite sur la position relative de ces antennes. Ainsi, elle peuvent être proches ou bien réparties dans l'espace, disposées selon une ligne, un cylindre, un rectangle ou aléatoirement par exemple. La figure 1.2 présente les trois géométries les plus communes en représentant une antenne du réseau par un triangle. Nous supposons également que le récepteur dispose d'un moyen d'estimer sans erreur le gain complexe du canal de transmission liant chaque antenne d'émission à chaque antenne de réception.

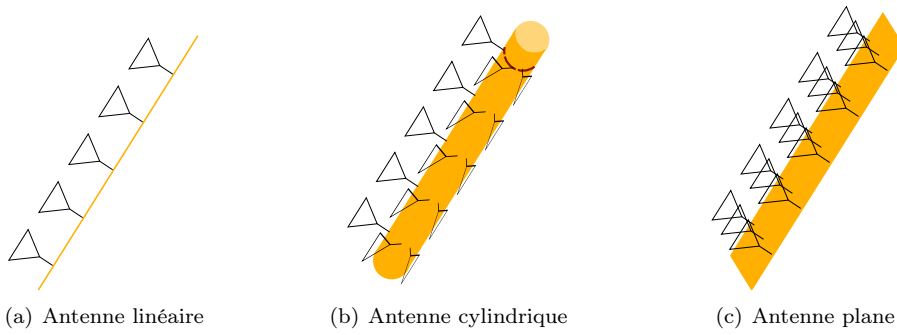


Figure 1.2) Les trois géométries les plus communes pour un réseau d'antennes MIMO.

A l'inverse, nous n'imposons aucune contrainte sur le ou les systèmes d'émission. Ainsi, les signaux peuvent être émis par un système unique disposant de plusieurs antennes ou par un ensemble, possiblement hétérogène, de systèmes disposant d'une ou plusieurs antennes chacun. Dans tous les cas, nous supposons que les émetteurs ne disposent d'aucune connaissance sur l'état du canal de transmission. Enfin, nous supposons que les émetteurs sont synchrones.

Dans le cas de plusieurs émetteurs indépendants, un système d'émission ne connaît pas les données émises par les autres systèmes. Ce cas de figure correspond par exemple à la liaison montante d'un système MIMO à utilisateurs multiples (multi-user MIMO, MU-MIMO). Dans ce format, il n'est pas possible de déployer un codage de canal commun à tous les flux de données. Ainsi, les résultats sans transmission codée de ce manuscrit seront applicables en l'état à ce cas de figure tandis que les résultats avec

## 1 Contexte, problématiques et hypothèses du manuscrit

codage de canal devraient être adaptés en utilisant un codage de canal par système d'émission.

Dans le cas d'un émetteur unique, tous les flux de données peuvent être codés par le même code de canal. C'est le cas de figure qui sert de base à toutes les simulations et tous les résultats de ce manuscrit s'y appliquent donc directement.

L'ensemble des simulations seront réalisées avec autant d'antennes d'émission que d'antennes de réception ce qui n'impose rien sur le nombre de système d'émission. Enfin, nous étudions uniquement le cas de transmissions sans codage spatio-temporel utilisant le multiplexage spatial pour augmenter le débit de données. Nous nous concentrons sur les systèmes MIMO conventionnels mais rien n'empêcherait de transposer les travaux présentés au MIMO massif.

### Standards entrants dans ce cadre général

Deux standards répandus entrent dans le cadre défini par les hypothèses précédentes. Nous les présentons ici rapidement pour guider les choix de modélisation qui suivront dans la fin de ce chapitre. L'ensemble des modulations et des codes correcteurs d'erreur cités ici seront détaillés dans la suite du chapitre.

Les deux standards implémentent des techniques de codages et modulations adaptatifs (adaptive coding and modulations, AMC). En d'autres mots, la modulation et le code correcteur d'erreur sont choisis en fonction de l'état du canal de transmission. Ainsi, une modulation simple et un codage fort seront utilisés lorsque le rapport signal à bruit (signal-to-noise ratio, SNR) est faible afin d'assurer une robustesse de transmission. A l'inverse, la recherche de débit conduit à choisir des modulations d'ordre supérieur et des codes simples dans les cas de canaux très faiblement bruités.

La Wi-Fi est un ensemble de protocoles défini par les standards IEEE 802.11 qui inclut l'utilisation des technologies MIMO [27, 28]. Elle permet le transport d'information en intérieur ou extérieur sur une distance allant de la dizaine de mètres en intérieur à une grosse centaines de mètres dans des conditions optimales. Le standard définit une dizaine de modulation and coding schemes (MCS) qui fixe chacun une modulation et un code correcteur d'erreur susceptibles d'être choisis dans le cadre des AMC. Les modulations standardisées sont toutes les modulations d'amplitude en quadrature (squared amplitude modulations, QAM) comprenant de 4 à 256 symboles ainsi que la modulation par changement de phase binaire (binary phase-shift keying, BPSK). La Wi-Fi utilise des codes convolutifs de rendement compris entre 1/2 et 5/6. Le tableau 1.1 liste les configurations prévues par la norme.

MCS	0	1	2	3	4	5	6	7	8	9
Nb symboles	2	4	4	16	16	64	64	64	256	256
Rendement code	1/2	1/2	3/4	1/2	3/4	2/3	3/4	5/6	3/4	5/6

Tableau 1.1) Nombre de symboles dans la constellation et rendement du code convolutif en fonction du MCS selon le standard Wi-Fi/IEEE 802.11.

Le WiMAX est un ensemble de protocoles basé sur les standards IEEE 802.16 qui

rassemble les protocoles dédiés à la transmission hertzienne moyenne et longue portée au sein de la famille des standards IEEE 802 [29]. Il fournit une alternative aux transmissions câblées pour connecter des utilisateurs au cœur du réseau en l'absence d'infrastructure suffisante. Il peut aussi être déployé en doublon d'un réseau câblé pour assurer la continuité de service en cas d'accident. Le WiMAX standardise les différentes utilisations possibles du MIMO : l'augmentation du débit et/ou de la fiabilité. Ainsi, le *mode A* utilise plusieurs antennes et un code spatio-temporel pour augmenter la fiabilité à débit constant. À l'inverse, le *mode C* utilise un système 4x4 pour multiplier le débit par 4 à fiabilité constante. Enfin, le *mode B* mixe les deux en multipliant le débit par 2 et en augmentant un peu la fiabilité. Dans tous les modes, le WiMAX s'appuie sur des QAM ainsi que sur un code à contrôle de parité de faible densité (low-density parity-check, LDPC).

### 1.2.2 Canal de transmission

La figure 1.3 schématise les multi-trajets et les interférences caractéristiques des systèmes MIMO. La figure prend l'exemple la liaison montante d'un système MU-MIMO et omet certaines réflexions sur les obstacles par lisibilité. Dans un cas réel, chaque flux de données subit plusieurs multi-trajets en plus de l'éventuelle ligne directe. La présence et l'impact de ligne directe est détaillée par la suite.

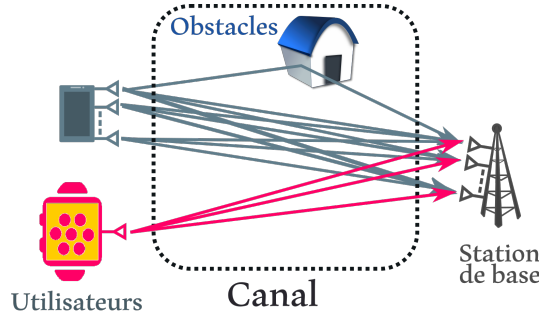


Figure 1.3) Multi-trajets lors d'une liaison montante d'un système MIMO.

Le bloc *canal de transmission* agrège tous les phénomènes qui altèrent la transmission des symboles entre l'émetteur et le récepteur. Soit  $N_t$  le nombre d'antennes de l'émetteur et  $N_r$  celui du récepteur. Le canal est alors représenté mathématiquement par une matrice complexe  $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$  où le coefficient  $h_{i,j}$  correspond au gain complexe du canal entre l'antenne émettrice  $j$  et l'antenne réceptrice  $i$ . Lorsque le canal possède autant d'antennes d'émission que de réception, nous notons  $N \triangleq N_t = N_r$ .

Le canal est considéré normalisé de telle sorte qu'il ne puisse ni ajouter, ni retirer d'énergie. Cette propriété se traduit par

$$\|\mathbf{H}\|_F^2 = \text{Tr}(\mathbf{H}\mathbf{H}^H) = N_t N_r \quad (1.1)$$

avec  $\cdot^H$  la transposition hermitienne et  $\|\cdot\|_F$  la norme de Frobenius [30]. Le canal

de transmission intègre également la somme de tous les bruits (circuits analogiques, conversion analogique-numérique et bruits numériques) sous la forme d'un vecteur  $\mathbf{w} \in \mathbb{C}^{N_r}$  dont chaque coordonnée  $w_i \sim \mathcal{CN}(0, \sigma^2)$  suit une loi normale complexe centrée de variance  $\sigma^2$  à symétrie circulaire.

Le canal est traditionnellement considéré comme une matrice aléatoire restant stable pendant un temps dénommé *temps de cohérence*. Dans des conditions normales, ce temps est suffisamment long pour transmettre plusieurs centaines de symboles [31]. Une transmission MIMO classique consiste donc en une estimation du canal à partir de séquences pilotes puis en la transmission d'un maximum de symboles avant que le canal ne change. Dans ce modèle, la question de la modélisation du canal revient à spécifier la loi de probabilité de la matrice aléatoire du canal.

### Canal de Rayleigh non corrélé

Le modèle le plus simple, est le canal de Rayleigh non corrélé dans lequel chaque coefficient de la matrice de canal  $H$  suit une distribution normale centrée réduite complexe à symétrie circulaire. Cela s'exprime mathématiquement par

$$\forall i \in \llbracket 1, N_r \rrbracket, \forall j \in \llbracket 1, N_t \rrbracket, h_{i,j} \sim \mathcal{CN}(0, 1). \quad (1.2)$$

On parle de canal non corrélé quand les réalisations des variables aléatoires de (1.2) sont toutes indépendantes ce qui peut être valide ou pas selon la position relative des antennes.

Le modèle de Rayleigh représente bien un canal de transmission riche en réflexions dans lequel il n'y a pas de visibilité directe entre émetteur et récepteur. En effet, le signal reçu est alors la somme d'un très grand nombre de réflexions de telle sorte que le théorème central limite puisse s'appliquer. Le canal de Rayleigh non corrélé est largement utilisé pour sa simplicité qui permet d'obtenir des résultats analytiques. Il peut être raffiné par l'ajout d'une composante déterministe correspondant à un trajet direct entre émetteur et récepteur ce qui donne le canal de Rice et/ou par l'ajout d'une corrélation entre les gains complexes ce qui donne un canal corrélé.

### Le canal de Rice non corrélé

Lorsqu'il existe un trajet direct entre l'émetteur et le récepteur, une composante déterministe correspondant au trajet direct (noté LOS pour *line of sight*) s'ajoute à la composante aléatoire (noté NLOS pour *non line of sight*) précédemment décrite [32]. On parle alors de *canal de Rice*. Mathématiquement, on décompose la matrice de canal en une composante  $\mathbf{H}^{\text{LOS}}$  de ligne directe et une composante  $\mathbf{H}^{\text{NLOS}}$  de réflexion selon la forme

$$\mathbf{H} = \sqrt{\frac{K}{K+1}} \mathbf{H}^{\text{LOS}} + \sqrt{\frac{1}{K+1}} \mathbf{H}^{\text{NLOS}} \quad (1.3)$$

où  $K$  est le rapport de puissance entre ligne de vue direct (line-of-sight, LOS) et sans ligne de vue (non-line-of-sight, NLOS) [25]. Le terme  $\mathbf{H}^{\text{NLOS}}$  correspond au canal de Rayleigh précédemment décrit et le terme déterministe dépend de la forme du réseau d'antenne et de l'environnement.

### Canal corrélé : modèle de Kronecker et corrélation exponentielle

Les antennes d'un réseau étant très proches les unes des autres, il est possible de considérer que le canal observé par une antenne sera corrélé à celui observé pour l'antenne voisine. Le modèle le plus classiquement utilisé est le modèle de corrélation de Kronecker qui permet de scinder la matrice de corrélation du canal  $\mathbf{R}$  en une matrice de corrélation à l'émission  $\mathbf{R}_T$  et une à la réception  $\mathbf{R}_R$ . On a alors

$$\mathbf{R} \triangleq \mathbf{R}_T^T \otimes \mathbf{R}_R \quad (1.4)$$

avec  $\otimes$  le produit de Kronecker qui donne son nom au modèle [33]. Ainsi, la matrice de corrélation du canal est définie par bloc :

$$\mathbf{R} \triangleq \begin{pmatrix} r_{T,1,1} \mathbf{R}_R & \cdots & r_{T,N_t,1} \mathbf{R}_R \\ \vdots & \ddots & \vdots \\ r_{T,1,N_t} \mathbf{R}_R & \cdots & r_{T,N_t,N_t} \mathbf{R}_R \end{pmatrix} \quad (1.5)$$

avec  $r_{T,i,j}$  le coefficient  $(i,j)$  de la matrice  $\mathbf{R}_T$ . La matrice de canal peut alors être obtenue à partir d'une matrice non corrélée  $\mathbf{H}_N$  issue de l'un des modèles précédent et de l'expression :

$$\mathbf{H} = \mathbf{R}_R^{\frac{1}{2}} \mathbf{H}_N \mathbf{R}_T^{\frac{t}{2}} \quad (1.6)$$

avec  $\cdot^{\frac{1}{2}}$  la racine carrée matricielle et  $\cdot^{\frac{t}{2}}$  celle de la transposée [30]. Ces deux racines matricielles sont bien définies car toute matrice de corrélation est demi-définie positive et cette propriété est conservée par transposition. Le modèle de Kronecker peut donc ajouter une corrélation aux modèles de Rayleigh et de Rice.

On utilise classiquement dans le modèle de corrélation exponentielle des antennes linéaires qui expriment que la corrélation décroît exponentiellement avec la distance entre les antennes. Mathématiquement, le coefficient  $r_{i,j}$  d'une matrice de corrélation s'écrit alors

$$r_{i,j} \triangleq t^{i-j} \quad (1.7)$$

avec  $t$  un complexe, tel que  $|t| \leq 1$ , décrivant l'influence plus ou moins grande de la corrélation [34, 35].

Bien qu'il s'agisse d'une hypothèse, il a été montré que le modèle de Kronecker fournit des résultats similaires à des modèles plus évolués [33]. De plus, les mesures de canaux en intérieur correspondent bien à un canal de Rayleigh corrélé selon ce modèle [36]. Enfin, cette hypothèse permet de simplifier le phénomène de corrélation pour permettre de réaliser des calculs analytiques [37, 38]

### Conclusions et hypothèses du manuscrit

Nous avons résumé dans cette section les différents modèles élémentaires de canaux MIMO. Ces modèles servent de base pour la plupart des modèles utilisés dans la littérature ainsi que pour les modèles standardisés sous le nom TGn par IEEE [39].

Dans la suite du manuscrit nous utiliserons des canaux de Rayleigh non corrélés sauf mention contraire. Ce choix est d'abord soutenu par la simplicité de ce type de

canaux qui a toujours servi de première étape dans la littérature. De plus, des mesures en extérieur avec des antennes linéaires et cylindriques suggèrent que les canaux réels tendent vers des canaux de Rayleigh non corrélés quand le nombre d’antennes augmente significativement [40]. Enfin, nous nous fixons, pour les simulations, un canal 4x4 en accord avec les cas d’utilisation Wi-Fi et WiMAX présenté en section 1.2.1.

### 1.2.3 Modulation et démodulation

La modulation numérique est le bloc qui associe à un mot binaire de  $m$  bits un symbole complexe choisi parmi les  $2^m$  disponibles dans une constellation. Sauf mention contraire, ce manuscrit restreint l’étude aux QAM carrées, ce qui représente la grande majorité des standards usuels (LTE, Wi-Fi, WiMAX, 4G et cinquième génération (5G) mobiles *etc*). Dans un système MIMO, le train binaire à transmettre est modulé pour obtenir les symboles puis un symbole est envoyé à chaque antenne d’émission.

La démodulation peut prendre deux formes différentes dépendant du type de détection utilisé. Dans le cas d’une détection à sortie dure (*hard-output*), le démodulateur reçoit une série de symboles de la constellation pour la traduire en un message binaire. Il s’agit alors simplement d’appliquer la fonction réciproque de la fonction de modulation. Les mots binaires de sortie peuvent alors être lus directement ou transmis au décodeur. Dans le cas d’une détection à sortie souple (*soft-output*) la démodulation est incluse dans la détection afin que le bloc détection-démodulation produise une sortie souple compatible avec le décodeur. Cette sortie souple prend la forme du ratio des log-vraisemblances (log-likelihood ratio, LLR) de chaque bit  $b$  défini par

$$L \triangleq \ln \frac{P(b = 1 | (\mathbf{y}, \mathbf{H}))}{P(b = 0 | (\mathbf{y}, \mathbf{H}))} \quad (1.8)$$

avec  $P(\cdot | (\mathbf{y}, \mathbf{H}))$  la probabilité conditionnelle sachant l’information reçue sur chaque antenne et l’état du canal.

### 1.2.4 Codage et décodage

Le codage de canal est une étape qui réduit l’influence des erreurs de transmission sur l’intégrité du message. Elle est théoriquement optionnelle mais la quasi-totalité des systèmes modernes l’implémentent. Pour réduire l’influence des erreurs, le décodeur exploite la redondance ajoutée dans le message émis pour localiser et corriger les erreurs.

La fonction de l’encodeur est de prendre un message binaire et d’ajouter une redondance selon un procédé donné afin de produire un nouveau message binaire plus long mais contenant exactement la même quantité d’information. Ce procédé peut appliquer en continu de la redondance à un flux binaire ou bien protéger le message bloc par bloc. Comme pour la démodulation, deux types de décodeurs existent selon que la détection soit à sortie dure ou souple. Dans le cas d’une détection à sortie dure, le décodeur travaille sur le message binaire détecté et en corrige les erreurs selon un algorithme donné. Lorsque la détection est souple, les entrées du détecteur sont les LLR de chaque bit plutôt que le message binaire [41].



## 1.2 Décomposition d'une liaison MIMO

Nous introduisons succinctement dans la suite les deux couples codeur-décodeur qui seront utilisés dans les simulations de ce manuscrit.

### Code convolutif non récursif - Décodeur de Viterbi

Les codes convolutifs non récursifs ajoutent en continu des bits de redondance à un flux de bits d'information en le convoluant avec le polynôme générateur du code. La convolution implique que les codes convolutifs sont des codes à mémoire où tous les bits précédemment codés ont une influence sur la suite. Le décodage des codes convolutifs repose classiquement sur l'algorithme de Viterbi qui fournit des performances optimales tout en étant très parallélisable.

Dans ce manuscrit, le code convolutif généré par  $(1 + x + x^2, 1 + x^2)$  (soit sous forme octale (7, 5)) sera utilisé pour les études préliminaires en raison de sa simplicité. Ce code n'a pas vocation à représenter une utilisation réelle mais permet une étude simplifiée de la chaîne de transmission. Pour autant, ce code convolutif de rendement 1/2 est proche des codes présents dans la Wi-Fi à la différence que la version utilisée ici est beaucoup plus simpliste. La figure 1.4 représente ce code sous deux formes : un diagramme des registres montrant l'encodeur et une machine à états décrivant les transitions et les sorties correspondantes entre parenthèse.

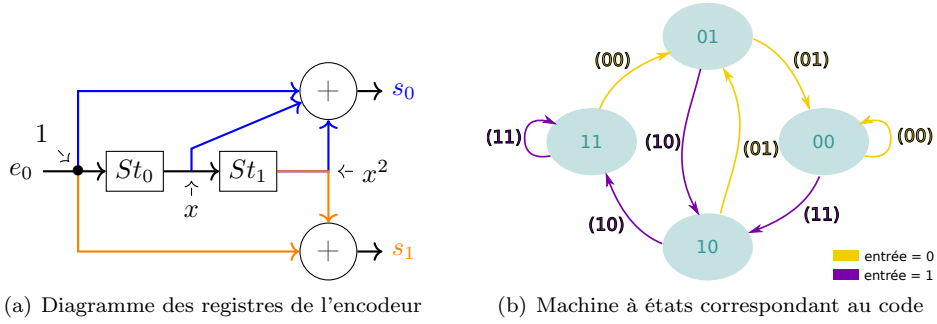


Figure 1.4) Représentations du code convolutif généré par  $(1 + x + x^2, 1 + x^2)$

### LDPC - Décodeur par propagation des convictions (belief propagation, BP)

Les codes LDPC sont des codes linéaires en bloc très performants utilisés dans plusieurs standards tel que le Wi-Fi ou le WiMAX ou encore pour la protection du champs de données dans la 5G mobile. Les LDPC sont construits à partir de graphes de Tanner clairsemés pour bénéficier de blocs longs tout en conservant une complexité de décodage acceptable. Le message codé est classiquement décodé itérativement par un algorithme par BP avec plusieurs variantes. Un code à LDPC est caractérisé par sa matrice de contrôle qui représente les connections dans le graphe de Tanner, qui permet de détecter les erreurs et de les corriger conjointement à l'algorithme par BP.

## 1 Contexte, problématiques et hypothèses du manuscrit

Les simulations plus proches des conditions réelles de ce manuscrit seront menées en utilisant un code à LDPC irrégulier, systématique. La matrice de contrôle sera construite pour des blocs de 720 bits pour mots codés de 1440 bits (*ie*, un rendement de 1/2) suivant la norme WiMAX [29]. Ainsi, en utilisant ce code avec une QAM et un canal 4x4, les simulations reprendront complètement les paramètres du WiMAX *mode C*. La matrice de contrôle est représentée sur la figure 1.5 qui met en évidence la parcimonie de la matrice avec 99.6% de zéros.

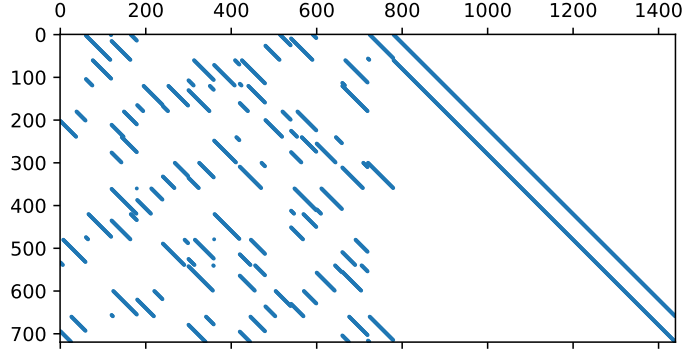


Figure 1.5) Coefficients non nuls dans la matrice de contrôle du code LDPC 720x1440 issu de la norme WiMAX

### 1.2.5 Synthèse de la modélisation mathématique

La figure 1.6 reprend la décomposition de la figure 1.1 en introduisant toutes les notations mathématiques. On note  $\mathbf{m}$  le message de  $L_m$  bits que l'on souhaite transmettre. Ce message est potentiellement codé en un message  $\mathbf{b}$  de  $L_c$  bits avec un rendement de  $L_m/L_c$ . Dans le cas d'une transmission sans codage, on a  $\mathbf{b} = \mathbf{m}$ . Le message binaire est traduit avec le  $q$  symboles de la constellation  $\mathcal{Q}$  selon le ratio de  $\log_2 q$  bits par symbole. Les symboles sont séparés en  $N_t$  flux de données et on note  $\mathbf{x}$  le vecteur comprenant les symboles émis sur chaque antenne. Après application de la matrice du canal  $\mathbf{H}$  et ajout du vecteur de bruit  $\mathbf{w}$  de variance  $\sigma^2$ , le récepteur récupère le vecteur complexe  $\mathbf{y}$ . Après détection, le récepteur réalise une estimation des symboles émis  $\hat{\mathbf{x}}$  ou les LLR de chaque bit  $\mathbf{L}$ . Enfin, la démodulation et le décodage fournissent les estimations des messages binaires codés  $\hat{\mathbf{b}}$  et décodés  $\hat{\mathbf{m}}$ .

Avec toutes ces notations, on peut écrire le message reçu comme :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}. \quad (1.9)$$

Les chapitres 3 et 4 de ce manuscrit redéfinissent ce modèle avec des variables réelles. Cette nouvelle version est introduite dans la section 3.1.1.

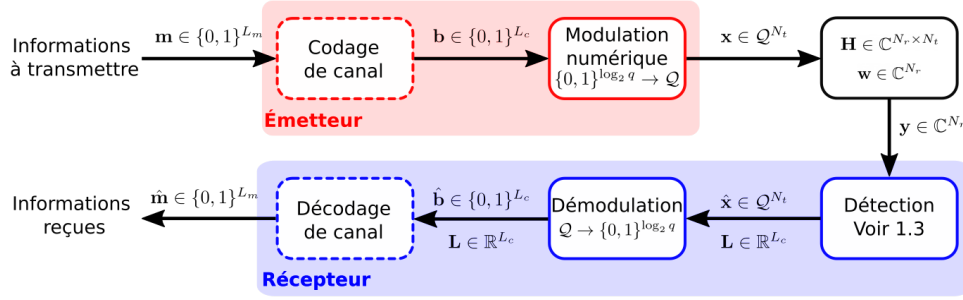


Figure 1.6) Modélisation mathématique de la chaîne de transmission considérée.

### 1.3 État de l'art des détecteurs MIMO

Le détecteur a pour fonction de séparer les différents flux de données et de compenser au mieux les effets du canal de transmission en exploitant les informations reçues sur chaque antenne et l'estimation du canal de transmission. Mathématiquement, le problème de détection fait référence au problème d'optimisation combinatoire

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{Q}^{N_t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \quad (1.10)$$

La résolution du problème (1.10) équivaut à rechercher le vecteur  $\hat{\mathbf{x}}$  le plus probablement émis en fonction du vecteur  $\mathbf{y}$  et de l'état du canal sous l'hypothèse d'un bruit gaussien à symétrie circulaire et de symboles équiprobables. Il faut noter que même si  $E(\mathbf{x}) \triangleq \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$  est une fonction convexe par rapport à  $\mathbf{x} \in \mathbb{C}^{N_t}$ , le problème de détection n'est pas un problème d'optimisation convexe puisque l'ensemble des possibles  $\mathcal{Q}^{N_t}$  est discret. Nous appellerons dans toute la suite de ce manuscrit *fonction objectif* la fonction  $E$ .

Cette section liste les principaux algorithmes de détection MIMO en soulignant les différents paradigmes utilisés. La section 1.3.1 présente le détecteur optimal maximum de vraisemblance (maximum likelihood, ML), puis la section 1.3.2 décrit les détecteurs linéaires. La section 1.3.3 détaille les algorithmes basés sur l'annulation des interférences et la section 1.3.4 ceux basés sur des arborescences. Enfin, la section 1.3.5 présente des détecteurs par réseau de neurones, la section 1.3.6 s'intéresse aux méta-heuristiques bio-inspirées et la section 1.3.7 introduit les détecteurs géométriques. La figure 1.7 classe tous les détecteurs décrits dans ce chapitre et donne le numéro de la section correspondante. Cette section se veut assez détaillée pour mettre en lumière les caractéristiques qui serviront la synthèse dans la section 1.4. L'ensemble de cet état de l'art s'appuie sur le chapitre que nous avons publié dans [1].

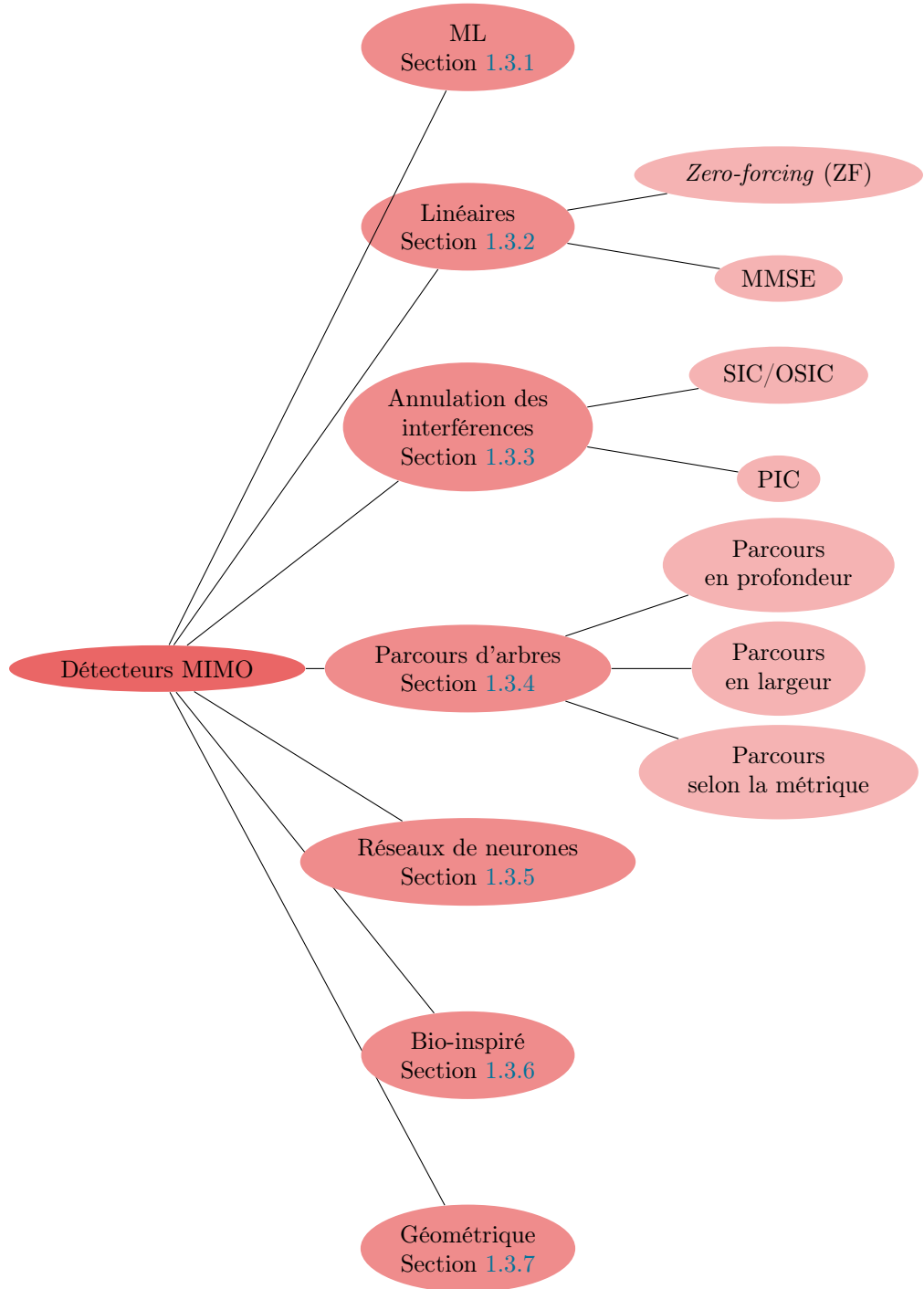


Figure 1.7) Taxonomie des détecteurs MIMO et numéro de la section qui lui est dédiée

### 1.3.1 Détecteur par maximum de vraisemblance

L'approche naïve pour résoudre le problème (1.10) est de réaliser une recherche exhaustive, ce qui fournit un résultat optimal au sens du maximum de vraisemblance (maximum likelihood, ML) puisque tous les possibles de  $\mathbf{x} \in \mathcal{Q}^{N_t}$  sont évalués. Pour autant, cette méthode n'est, bien sûr, pas utilisable en pratique en raison de son coût algorithmique prohibitif. En effet, le nombre de vecteurs à tester croît de manière exponentielle avec le nombre d'antennes d'émission et la taille de la constellation. Le cas relativement simple qu'une 16-QAM transmise dans un canal 4x4 requiert, par exemple, 65536 évaluations de fonction objectif, ce qui est prohibitif. C'est pour cette raison que de nombreux détecteurs ont été développés pour obtenir les mêmes performances de détection que les détecteurs ML tout en ayant une complexité acceptable.

Du point de vue de la théorie de la complexité, le problème de détection est une application du problème de recherche du plus proche point dans le réseau défini par  $\{\mathbf{H}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^{N_t}\}$  [42]. Il a été prouvé que, pour tout prétraitement sur le réseau de points (ie, la matrice de canal  $\mathbf{H}$ ), le problème reste NP-hard [42]. Il n'est donc pas possible à l'heure actuelle de trouver un détecteur qui ait à la fois une performance optimale et une complexité polynomiale<sup>1</sup>. Pour cette raison, tous les détecteurs qui seront présentés par la suite ont soit des performances sous-optimales (qui peuvent être très proches de l'optimum), soit une complexité non polynomiale dans le pire des cas (qui peut être raisonnable pour des systèmes MIMO conventionnels) [43].

### 1.3.2 Détecteurs linéaires

#### Détecteur par zero-forcing

La détection linéaire est l'approche la plus simple pour résoudre le problème de détection et le détecteur forçage à zéro (zero-forcing, ZF) est le plus simple dans cette classe d'algorithmes. Il procède en deux étapes successives.

Tout d'abord, la contrainte de (1.10) est relaxée en remplaçant  $\mathbf{x} \in \mathcal{Q}^{N_t}$  par  $\mathbf{x} \in \mathbb{C}^{N_t}$  afin que le problème devienne une optimisation convexe usuelle. La solution mathématique est alors bien connue :

$$\mathbf{x}_0 = \mathbf{H}^+ \mathbf{y} \quad (1.11)$$

avec  $\mathbf{H}^+ = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$  étant la pseudo-inverse de Moore-Penrose à gauche.

Ensuite, la contrainte sur  $\mathbf{x}$  est réintroduite en quantifiant le vecteur selon la constellation utilisée, c'est à dire en prenant le point de la constellation qui est le plus proche de  $\mathbf{x}_0$ . Cette quantification fournit une assez bonne estimation car l'équation (1.9) devient

$$\mathbf{T}_{ZF} \cdot \mathbf{y} = \mathbf{x} + \mathbf{H}^+ \mathbf{w} \quad (1.12)$$

<sup>1</sup>Il est fortement pressenti qu'il n'existe aucune solution pour résoudre les problèmes NP-hards en temps polynomial sur une machine de Turing déterministe. Cependant, cette hypothèse n'est pas prouvée et il est donc toujours possible qu'un tel algorithme existe et qu'il n'a tout simplement pas encore été découvert.

après application de la matrice de détection  $\mathbf{T}_{ZF} \triangleq \mathbf{H}^+$ . Cela souligne que toutes les interférences sont annulées.

L'équation (1.12) prouve également que le détecteur ZF est le détecteur linéaire optimal en ce qui concerne le rapport signal à interférences (signal-to-interferences ratio, SIR). En effet, on peut voir que le vecteur  $\mathbf{T}_{ZF} \cdot \mathbf{y}$  contient chaque flux séparé ainsi qu'un bruit coloré, mais qu'aucune interférence inter-flux n'est présente. Enfin, ce détecteur peut fournir un résultat optimal pour les matrices particulièrement bien conditionnées, *ie* telle que  $\lambda_{\max}/\lambda_{\min} \approx 1$  avec  $\lambda_{\max}$  la plus grande valeur singulière et  $\lambda_{\min}$  la plus petite.

### Détecteur à erreur quadratique moyenne minimale (MMSE)

La détection ZF se concentre exclusivement sur l'annulation des interférences pour séparer les flux de données les uns des autres mais cela se fait au détriment d'une modification du bruit. En effet, le terme  $\mathbf{H}^+ \mathbf{w}$  de (1.12) représente une coloration du bruit et une potentielle augmentation de son amplitude ce qui porte préjudice aux performances de détection.

Quand le récepteur connaît la variance du bruit, un estimateur bayésien incluant cette information peut fournir une meilleure détection. Le principe d'orthogonalité [44] permet ainsi de construire un estimateur bayésien linéaire minimisant l'erreur quadratique moyenne. Un tel détecteur est appelé détecteur à erreur quadratique moyenne minimale (minimum mean square error, MMSE) et s'exprime par :

$$\mathbf{T}_{MMSE} \triangleq \left( \mathbf{H}^H \mathbf{H} + \sigma^2 \mathbf{I} \right)^{-1} \mathbf{H}^H. \quad (1.13)$$

Lorsque le SNR est faible (*ie*  $\sigma^2$  est élevé), le détecteur MMSE fournit de meilleurs résultats en minimisant conjointement interférences et bruits. Il est alors le détecteur linéaire optimal pour minimiser le rapport signal à interférences plus bruit (signal-to-interferences-plus-noise ratio, SINR). Dans le cas contraire, (*ie*  $\sigma^2$  est très faible) le terme correctif devient négligeable, et les détecteurs ZF et MMSE se recouvrent.

### 1.3.3 Détecteurs à annulation des interférences

Il est nécessaire de complexifier les algorithmes de détection pour diminuer encore le taux d'erreurs binaires (bit error ratio, BER). Historiquement, la première classe de détecteurs non linéaires s'appuie sur l'annulation des interférences entre flux de données. Ce paradigme se décline en deux approches : annulation successive des interférences (successive interferences cancellation, SIC) et annulation des interférences en parallèle (parallel interferences cancellation, PIC).

#### Détection par annulation successive des interférences (SIC)

La détection SIC repose sur l'alternance de deux étapes jusqu'à l'obtention du résultat final : une décision sur un symbole de  $\mathbf{x}$  et une suppression des interférences liées à ce symbole dans  $\mathbf{y}$  en supposant que l'hypothèse était la bonne.

### 1.3 État de l'art des détecteurs MIMO

Les performances des détecteurs SIC sont bien meilleures que celles des algorithmes linéaires mais le processus de détection reste vulnérable à la propagation des erreurs. En effet, puisque l'hypothèse d'une itération a des répercussions sur toutes les itérations suivantes, une erreur en début de détection peut provoquer un effet boule de neige. Le détecteur SIC simpliste a donc été remplacé par une variante recherchant un ordre d'itération optimal [45]. Avec cette variante appelée annulation successive ordonnée des interférences (ordered successive interferences cancellation, OSIC), les hypothèses sont réalisées dans l'ordre qui maximise le SNR ou le SINR.

Ainsi, le détecteur OSIC évalue le SNR ou le SINR post-itération pour chaque hypothèse possible, en supposant que l'interférence soit annulée à l'aide d'une matrice de détection  $\mathbf{T}$  pour sélectionner le meilleur symbole à détecter à chaque itération. La plupart du temps, la matrice de détection est  $\mathbf{T}_{ZF}$  ou  $\mathbf{T}_{MMSE}$ , possiblement mise à jour après chaque itération.

Le SNR après l'annulation du  $k^{\text{ème}}$  symbole de  $\mathbf{x}$  est évaluée selon [45, 46] par

$$\text{SNR}_k = \frac{\mathbb{E}_{\mathcal{Q}_k}(x_k^2) |\mathbf{T}_k \mathbf{h}_k|^2}{\sigma^2 \|\mathbf{T}_k\|^2} \quad (1.14)$$

avec  $\mathbf{T}_k$  la  $k^{\text{ème}}$  ligne de  $\mathbf{T}$ ,  $\mathbf{h}_k$  la  $k^{\text{ème}}$  colonne de  $\mathbf{H}$  et  $\mathbb{E}_{\mathcal{Q}_k}(x_k^2)$  l'espérance de  $x_k^2$  sur l'ensemble de sa constellation  $\mathcal{Q}_k$ . Ce dernier terme correspond à la puissance moyenne du  $k^{\text{ème}}$  flux de données, ce qui revient avec des symboles équiprobables à

$$\mathbb{E}_{\mathcal{Q}_k}(x_k^2) \triangleq \frac{1}{q_k} \sum_{x \in \mathcal{Q}_k} x^2 \quad (1.15)$$

où  $q_k$  représente le nombre de symboles dans la constellation  $\mathcal{Q}_k$  utilisée pour le  $k^{\text{ème}}$  flux de donnée.

L'expression du SINR post-correction est un peu plus complexe car la puissance corrigée de chaque canal apparaît dans l'expression :

$$\text{SINR}_k = \frac{\mathbb{E}_{\mathcal{Q}_k}(x_k^2) |\mathbf{T}_k \mathbf{h}_k|^2}{\sum_{l \neq k} \mathbb{E}_{\mathcal{Q}_l}(x_l^2) |\mathbf{T}_l \mathbf{h}_l|^2 + \sigma^2 \|\mathbf{T}_k\|^2}. \quad (1.16)$$

Cette expression peut se simplifier si la puissance moyenne de chaque flux de données est identique, ce qui fait apparaître un terme proche du SNR :

$$\text{SINR}_k = \frac{|\mathbf{T}_k \mathbf{h}_k|^2}{\sum_{l \neq k} |\mathbf{T}_l \mathbf{h}_l|^2 + \frac{\sigma^2}{E_Q} \|\mathbf{T}_k\|^2} \quad (1.17)$$

avec  $E_Q$  la puissance moyenne de la constellation.

La figure 1.8 synthétise le processus de détection OSIC en introduisant une fonction  $t : \mathbf{H} \mapsto \mathbf{T}$  qui associe à une matrice de canal une matrice de correction. En règle générale, cette fonction construit  $\mathbf{T}_{ZF}$  ou de  $\mathbf{T}_{MMSE}$  telles que présentées dans la section 1.3.2. La figure introduit également l'ensemble des coordonnées encore à détecter  $\mathcal{D} = \{i \in \llbracket 1, N_t \rrbracket : \mathbf{x}_i \text{ n'est pas encore déterminé}\}$ .

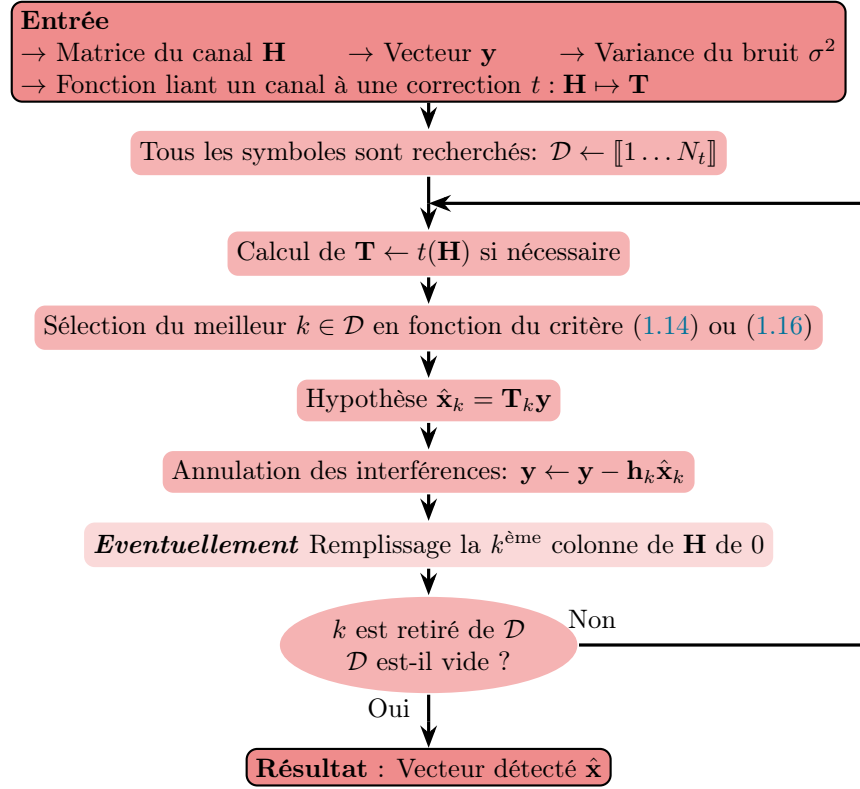


Figure 1.8) Résumé du détecteur OSIC

### Détection par annulation des interférences en parallèle (PIC)

Le principal inconvénient des détecteurs OSIC est que le nombre d'itérations croît linéairement avec le nombre d'antennes d'émission, ce qui devient un problème pour les systèmes MIMO de grande taille. C'est pourquoi, le principe de l'annulation des interférences a été adapté pour pouvoir traiter tous les symboles simultanément. C'est cette idée qui a mené au développement des détecteurs PIC.

La figure 1.9 décrit les étapes élémentaires de cette approche telle que publiée dans [47]. L'idée principale est d'utiliser un détecteur peu performant mais rapide à calculer, typiquement un détecteur linéaire, pour obtenir une première hypothèse puis d'annuler simultanément les interférences de tous les symboles à partir de celle-ci. En fonction des performances attendues, un fonctionnement itératif est possible pour raffiner le résultat en utilisant le résultat d'une itération comme hypothèse de la suivante.

Simultanément au développement des premiers détecteurs PIC, les techniques de réception itérative initialement développées pour le décodage des turbo-codes dans les systèmes à entrée et sortie unique (single-input single-output, SISO) sont adaptées



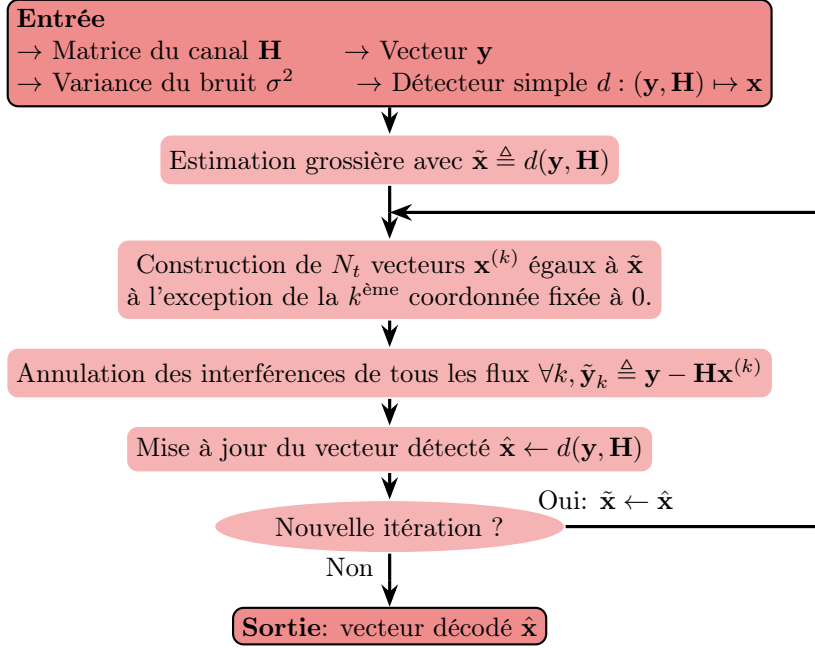


Figure 1.9) Algorithme PIC tel que publié dans [47]

pour les systèmes MIMO. L'objectif est ici de recevoir un message codé et d'augmenter les performances par des allers-retours entre détecteurs et décodeurs à entrée et sortie souple. Ainsi, chaque algorithme utilise l'information fournie par l'autre composant comme *a priori* de son propre calcul [48].

La version de base décrite dans la figure 1.9 a été largement améliorée pour produire des implémentations matérielles efficaces à entrée et sortie souples [49]. Tout d'abord, les symboles souples (*soft symbols*) permettent de traduire les *a priori* en symboles exploitables lors de l'obtention de l'estimation grossière [50]. De plus, des méthodes de réduction de complexité telles que l'approximation max-log [48, 51, 52] ou le recours à la matrice de Gram du canal [53] permettent une augmentation significative du débit pour des pertes de performances négligeables. Grâce à toutes ces améliorations, un circuit intégré spécifique à une application (*application-specific integrated circuit*, ASIC) a pu dépasser les 600 Mb/s de transmission codée pour un canal MIMO 4x4 et une 64-QAM [49].

### Choisir entre SIC et PIC

Le choix entre SIC et PIC s'appuie sur la qualité relative entre les différents flux de données. Comme cela a été noté précédemment, les détecteurs SIC et leurs variantes réalisent des hypothèses sur les symboles puis propagent ces postulats à travers l'annulation des interférences. Cette approche rend les détecteurs SIC très vulnérables

aux propagations d'erreur puisqu'une première hypothèse erronée engendre une erreur sur toute la suite du processus. Ainsi, ce type d'algorithme est bien adapté aux transmissions disposant de flux de données de qualité très différentes. En effet, les premières hypothèses seront très probablement valides car réalisées sur des flux de qualité et l'annulation de leurs interférences rendront la détection des flux de mauvaise qualité plus simple. Ce type de détection par ordre de qualité peut être rapproché de la détection réalisée dans les systèmes multiplexage non orthogonal (non-orthogonal multiple access, NOMA) opérant dans le domaine de puissance [54].

A l'inverse, les détecteurs PIC traitent tous les flux de données simultanément sans les discriminer sur leur qualité. Si cela permet une détection beaucoup plus rapide, cela signifie que la détection des flux de mauvaise qualité ne peut pas être aidée par les connaissances des flux de bonne qualité. Ainsi, ce type de détecteur est plus approprié pour traiter les cas où tous les flux sont de même qualité.

### 1.3.4 Détecteurs basés sur un parcours d'arborescence

Une grande partie des détecteurs développés ces dernières années utilise d'une façon ou d'une autre un parcours d'arborescence<sup>2</sup>. Dans ce paradigme, le problème de détection (1.10) est réinterprété de façon équivalente comme la recherche du meilleur chemin dans une arborescence pondérée. Il est important de noter que la théorie de la complexité a prouvé que ces deux approches sont mathématiquement équivalentes. Cette classe de détecteurs contient à la fois des détecteurs optimaux à complexité non polynomiale ainsi que des détecteurs à complexité polynomiale avec des résultats sous-optimaux.

La figure 1.10 donne un exemple de la réinterprétation du problème de détection par une arborescence avec deux flux de données et quatre symboles. Rappelons que résoudre le problème de détection (1.10) dans cette configuration revient à choisir les deux symboles de l'ensemble  $\mathcal{Q} = \{s_1, s_2, s_3, s_4\}$  qui minimise la fonction objectif. Or, cette recherche peut être considérée comme la recherche d'un chemin complet dans une arborescence pondérée dans lequel chaque niveau de l'arborescence correspond à une coordonnée de  $\hat{\mathbf{x}}$ . Les poids de chaque arête sont alors fixés de telle sorte que la métrique cumulée d'un chemin corresponde à la fonction objectif. L'affectation des poids requiert un développement mathématique intermédiaire et sera donc détaillée ultérieurement.

A titre d'illustration, la recherche exhaustive présentée en section 1.3.1 revient à calculer le poids cumulé de chaque chemin possible joignant le racine à une feuille puis à sélectionner le meilleur chemin. Pour autant, la structure en arborescence permet de faire appel à des algorithmes cherchant plus efficacement le meilleur chemin sans calculer la métrique cumulée de chaque chemin. On distingue trois grandes classes de parcours d'arborescence selon la façon de construire les branches : parcours en profondeur (depth-first search, DFS), parcours en largeur (breadth-first search, BFS) et parcours selon la métrique (metric-first search, MFS). La suite de cette section

---

<sup>2</sup>Une arborescence est un arbre (*ie*, un graphe acyclique orienté) possédant une unique racine et dont tous les nœuds sauf la racine ont un parent unique. La section 2.2 détaille le vocabulaire associé à cette structure.

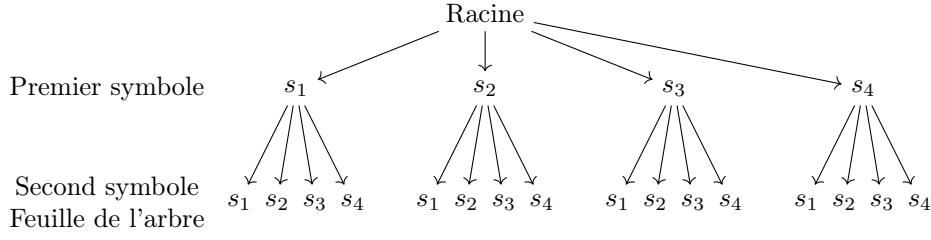


Figure 1.10) Représentation du problème de détection comme un parcours d'arborescence dans le cas de deux flux de données utilisant la constellation  $\mathcal{Q} = \{s_1, s_2, s_3, s_4\}$ .

décrit le prétraitement mathématique nécessaire à l'affectation des poids des arêtes puis présente les trois classes de parcours d'arborescence.

### Prétraitement par décomposition QR

Le prétraitement permettant d'affecter les poids à chaque arête du graphe reposant sur un parcours d'arborescence est commun à tous les détecteurs. Soit  $\mathbf{H} = \mathbf{Q}\mathbf{R}$  la décomposition QR de la matrice du canal avec  $\mathbf{Q}$  une matrice unitaire et  $\mathbf{R}$  une matrice triangulaire supérieure. Cette décomposition demeure valable pour toute la durée du bloc de cohérence de telle sorte que le surcoût est négligeable une fois rapporté au nombre de symboles détectés. Cette décomposition permet d'écrire

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\| = \|\mathbf{y} - \mathbf{Q}\mathbf{R}\mathbf{x}\| = \|\mathbf{Q}^H \mathbf{y} - \mathbf{Q}^H \mathbf{Q}\mathbf{R}\mathbf{x}\| \quad (1.18)$$

puisque les matrices unitaires décrivent des isométries. De plus, en exploitant la propriété des matrices unitaires  $\mathbf{Q}^H = \mathbf{Q}^{-1}$ , cette norme peut être réécrite comme

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\| = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\| \quad (1.19)$$

avec  $\tilde{\mathbf{y}} \triangleq \mathbf{Q}^H \mathbf{y}$ . Le calcul de  $\tilde{\mathbf{y}}$  est le seul surcoût du prétraitement à réaliser à chaque temps symbole.

Tout l'intérêt de ce prétraitement repose sur la triangularité de  $\mathbf{R}$ . En effet, il est alors possible de calculer par une somme cumulée la fonction objectif. Soit, pour tout  $k \in \llbracket 1, N_t \rrbracket$ ,  $d_k \triangleq \tilde{y}_k - (\mathbf{R}\mathbf{x})_k$  avec  $(\mathbf{R}\mathbf{x})_k$  le  $k^{\text{ème}}$  coefficient du produit  $\mathbf{R}\mathbf{x}$ . Cette définition permet de réécrire la fonction objectif comme

$$\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2 = \sum_{k=1}^n d_k^2. \quad (1.20)$$

De plus, la triangularité de  $\mathbf{R}$  implique que

$$\forall k \in \{1, \dots, N_t\}, (\mathbf{R}\mathbf{x})_k = \sum_{j=1}^n R_{k,j} x_j = \sum_{j=k}^n R_{k,j} x_j, \quad (1.21)$$

d'où

$$\forall k \in \{1, \dots, N_t\}, d_k = \tilde{y}_k - \sum_{j=k}^n R_{k,j} x_j. \quad (1.22)$$

Après ce prétraitement, il est possible de construire  $d_k$  ainsi que d'estimer la fonction objectif au fur et à mesure que le vecteur détecté est construit. En effet, à condition de traiter les coordonnées de  $\mathbf{x}$  de la dernière à la première, une hypothèse sur  $x_j$  détermine complètement la valeur de  $d_k$  à partir de (1.22). De plus, les nouveaux  $d_k$  peuvent être ajoutés à l'estimation de la norme selon (1.20) dès qu'ils sont disponibles. Ainsi, la construction itérative d'une distance euclidienne partielle (partial Euclidian distance, PED) permet d'estimer la qualité d'un chemin non terminé. Le poids de chaque arête de l'arborescence est donc fixé au  $d_k^2$  adéquat.

### Parcours d'arborescence en profondeur: *sphere decoding*

Le parcours en profondeur est le premier à avoir été introduit pour résoudre le problème de détection MIMO. Il s'agit en fait de l'application de l'algorithme mathématique de Fincke-Pohst [55] à la détection sous le nom de détection sphérique (sphere decoding, SD). Cette approche introduit un rayon maximal  $r$  pour élaguer le plus tôt possible des branches partielles afin d'éviter des constructions inutiles. En reprenant le  $\mathbf{x}_0$  défini dans (1.11), la contrainte de rayon se traduit par

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{H}(\mathbf{x}_0 - \mathbf{x})\|^2 \leq r^2, \quad (1.23)$$

ce qui revient à chercher une solution dans une hyper-sphère de rayon  $r$  centrée en  $\mathbf{x}_0$ .

L'équation (1.12) relevait que la seule déviation par rapport à  $\mathbf{x}_0$  provenait du bruit après application de  $\mathbf{T}_{ZF}$ . C'est pourquoi le choix du rayon doit être adapté au SNR : à SNR élevé, le rayon peut être faible tandis que dans le cas contraire, le rayon doit être plus important. Dans tous les cas, il est nécessaire de s'assurer qu'au moins un point de  $\mathcal{Q}^{N_t}$  satisfait la contrainte (1.23).

Deux approches sont possibles pour fixer le rayon. Tout d'abord, il est possible de déterminer en avance ce qui correspond à un parcours d'arbre avec retour sur trace (*backtrack*). Il est également envisageable de le maintenir à jour à la meilleure fonction objectif rencontrée jusqu'à présent, ce qui correspond à un algorithme de séparation et évaluation (branch and bound, BnB).

Dans la suite, nous supposons que le rayon  $r$  est choisi de façon adéquate et qu'il existe bien au moins un point de  $\mathcal{Q}^{N_t}$  dans l'hyper-sphère. Comme indiqué dans la section précédente, la décomposition QR permet de calculer une PED à chaque niveau de l'arborescence. Or, le poids de chaque arête de l'arbre est un carré, qui est donc positif par construction de telle sorte que la PED ne peut qu'augmenter lors de la construction d'un chemin. Ainsi, si un chemin partiel viole la contrainte (1.23) à un certain niveau, alors nous savons que tous les chemins construits à partir de celui-ci violeraient également cette contrainte. Il est donc possible d'élaguer tous les chemins partant de ce nœud sans aucune dégradation de performance tout en allégeant énormément la complexité du détecteur.

Le détecteur SD est appelé "à parcours en profondeur" car il part de la racine et cherche à aller le plus loin possible jusqu'à atteindre une feuille ou violer la contrainte de rayon. Si une feuille est atteinte, une mise à jour du meilleur chemin terminé est réalisée. Si la contrainte est violée, l'algorithme fait marche arrière et explore un nouveau chemin. Une fois tous les chemins complétés ou élagués, le résultat final est le meilleur chemin complet construit. La figure 1.11 propose une synthèse graphique du parcours d'arborescence en profondeur. Cette méthode de parcours garantit que le détecteur produise un résultat mathématiquement optimal. En effet, la meilleure feuille atteinte est, par construction, la meilleure feuille parmi tous les chemins complétés. De plus, les chemins élagués n'auraient pas pu mener à un meilleur chemin puisque leur PED était déjà plus grande que  $r$ . Comme le problème traité est NP-dur et que les résultats sont optimaux, la complexité de cette classe de détecteur est nécessairement non polynomiale. Pour autant, la complexité exponentielle peut être assez lente pour concurrencer les détecteurs polynomiaux dans certaines conditions favorables (SNR élevé, nombre d'antennes faible) [43].

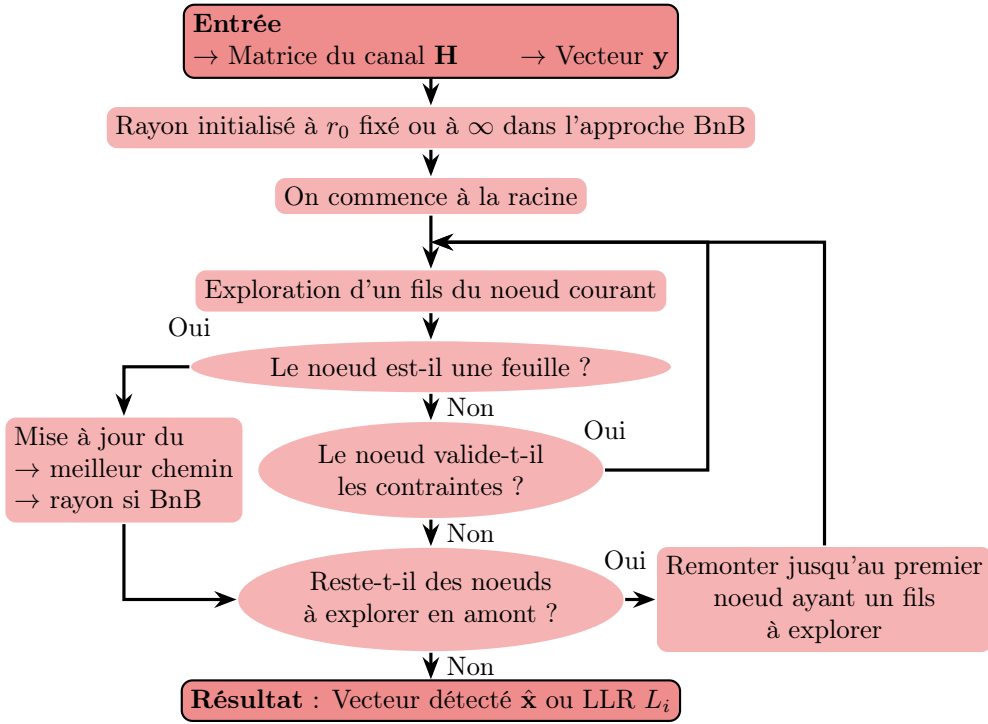


Figure 1.11) Résumé du parcours d'arborescence en profondeur

Des versions sous-optimales mais beaucoup plus rapides de détecteur à DFS ont également été proposées en s'appuyant sur les techniques de réduction de réseaux. Ces techniques visent à simplifier la base du réseau de points en une base quasi-orthogonale

pour simplifier les calculs. La figure 1.12 propose une illustration de ce principe en deux dimensions pour un réseau de points infini. Le même principe peut être utilisé pour simplifier la détection en réduisant le réseau engendré par la matrice de canal. Les utilisations les plus notables de ce principe utilisent l'énumération Schnorr-Euchner (SE) pour simplifier la détection [56, 57].

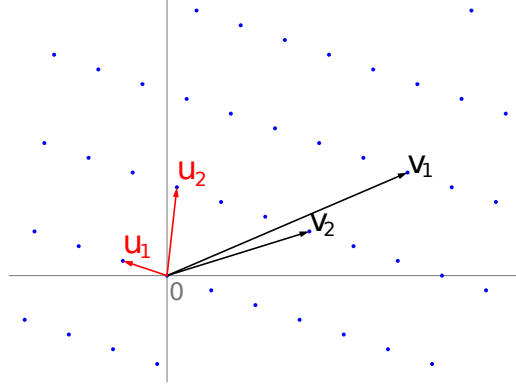


Figure 1.12) Réduction de réseau en deux dimensions : les vecteurs noirs représentent la base du réseau de points bleus et les vecteurs rouges représentent la base réduite. Les points bleus sont alors l'ensemble des points du réseau  $\{\mathbf{H}\mathbf{x} : \mathbf{x} \in \mathbb{Z}^2\}$  avec  $\mathbf{H}$  la matrice génératrice du réseau.

Des détecteurs SD à entrée et sortie souples ont aussi été développés comme le *single tree-search* par exemple [58]. Comme souvent, les LLR de (1.8) sont approximés avec l'approximation max-log [48, 51, 52] par

$$L_i \approx \frac{1}{2\sigma^2} \left( \min_{x \in \chi_i^0} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \min_{x \in \chi_i^1} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \right) \quad (1.24)$$

où  $\chi_i^k = \{x \in \mathcal{Q}^{N_t} : b_i = k\}$  est l'ensemble de tous les symboles dont le  $i^{\text{ème}}$  bit est fixé à  $k$ . L'expression de l'approximation met en évidence que deux minima différents doivent être trouvés pour obtenir une expression de  $L_i$  : le minimum global de la fonction objectif ainsi que la meilleure contre-hypothèse pour le  $i^{\text{ème}}$  bit. Ainsi, il faut s'assurer que les chemins pouvant améliorer la contre-hypothèse ne soient pas élagués. Pour ce faire, la PED est comparée au carré du rayon actuel  $r^2$  mais aussi à un nouveau rayon correspondant à la meilleur contre-hypothèse. Un chemin est donc élagué si sa PED viole simultanément les deux rayons.

Des implémentations sur ASIC des détecteurs DFS ont mis en évidence que ce paradigme permet d'atteindre de bons débits. L'implémentation de [59] dépasse, par exemple, les 600 Mb/s en sortie souple et les 1.2 Gb/s pour la version dure. Dans les deux cas, l'implémentation matérielle est testée sur un canal 4x4 et pour une 64-QAM. Elle exploite une énumération SE en deux dimensions et instancie plusieurs cœurs de détection en parallèle pour augmenter le débit.

### Parcours d'arborescence en largeur : K-Best et algorithme M

Les détecteurs basés sur un parcours en largeur (breadth-first search, BFS) abandonnent la recherche d'optimalité au profit d'une meilleure implémentabilité. En effet, cette approche permet de résoudre les deux freins majeurs de l'approche DFS à une implémentation efficace : la complexité variable dépendant du SNR et le retour sur trace. Pour ce faire, les détecteurs BFS comparent les PED à chaque niveau pour ne conserver qu'un nombre fixé de chemins. Ce nombre est traditionnellement appelé  $K$  pour l'algorithme K-best [3] ou  $M$  pour l'algorithme M [60]. Les travaux plus récents additionnent l'élagage dû à l'approche BFS avec l'élagage selon un rayon présenté précédemment pour réduire d'autant plus le nombre de chemins totaux construits [61].

L'élagage des moins bons chemins à chaque niveau implique que la solution du problème (1.10) peut être élaguée par erreur si sa PED a cru trop vite dans les premiers niveaux. Cet élagage erroné engendre bien évidemment une perte de performance de telle sorte que le nombre de chemins conservés à chaque niveau doit être suffisamment large pour éviter au maximum cette situation. Une autre solution pour limiter l'impact de ce problème est de réorganiser l'arborescence de façon à ce que les bons chemins soient les meilleurs dès les premiers étages. Les critères de SNR ou SINR post-détection définis en section 1.3.3 sont ainsi exploités pour obtenir un nouvel ordre plus favorable. Grâce à ces méthodes, l'algorithme K-Best, éventuellement couplé à une énumération SE atteignent des performances proches de l'optimum mathématique.

L'implémentation matérielle est le principal atout des détecteurs BFS car elle ne nécessite pas de retour arrière. Ainsi, un nœud traité pour obtenir ses enfants peut être supprimé définitivement. De plus, le nombre de nœuds visités à chaque niveau est fixé ce qui permet de prévoir précisément le nombre de ressources matérielles requises. Ces deux caractéristiques sont aussi déterminantes pour construire des pipelines matériels efficaces en termes de débit ou d'efficacité énergétique. Cette dernière est quantifiée par la quantité d'énergie consommée par bit d'information détecté. La figure 1.13 synthétise les différentes étapes de l'approche BFS et met bien en évidence la simplicité de ce dernier comparé à l'approche DFS de la section précédente. Une implémentation sur ASIC capable de s'adapter à la modulation et à la taille du canal a par exemple dépassé les 2.5 Gb/s [62]. Un autre exemple de la bonne implémentabilité est portée par une implémentation tournée sur l'économie d'énergie capable de corriger à la fois les erreurs dues au canal et celles dues à une sous-alimentation du circuit imprimé [63].

Les détecteurs à BFS peuvent fournir une sortie souple en associant l'approximation max-log à une liste de points. Pour ce faire, l'algorithme produit une liste de points  $\mathcal{S} \subset \mathcal{Q}^{N_t}$  associés à leur fonction objectif et approximée (1.24) par

$$L_i \approx \frac{1}{2\sigma^2} \left( \min_{x \in \mathcal{S} \cap \chi_i^0} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \min_{x \in \mathcal{S} \cap \chi_i^1} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \right). \quad (1.25)$$

Si les deux minima recherchés sont présents dans  $\mathcal{S}$ , alors cette version est équivalente à (1.24). Dans le cas contraire, les LLR sont bien approximés pour peu que le minimum diffère peu du minimum général. Les K-best usuels utilisent la liste des chemins complétés pour former la liste  $\mathcal{S}$  [3, 62].

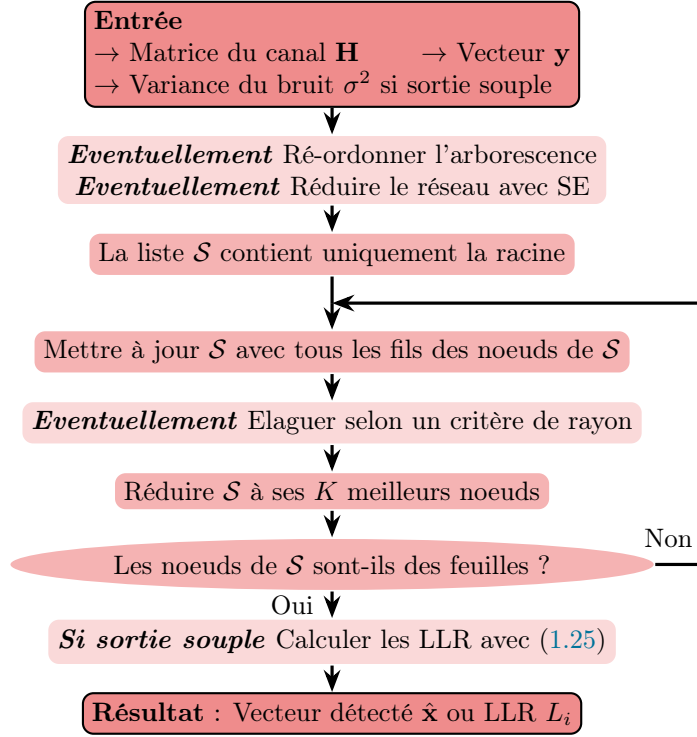


Figure 1.13) Résumé du parcours d'arborescence en largeur avec  $K$  le nombre de nœuds conservés par niveau.

### Parcours d'arborescence par métrique : *best-first*

Les détecteurs à parcours selon la métrique (metric-first search, MFS) sont la transcription de l'algorithme de Dijkstra [64] au domaine des télécommunications. Les algorithmes à MFS étendent en premier les chemins de plus faible PED plutôt que de privilégier la profondeur ou la largeur. A l'initialisation, la liste de chemins à étendre ne contient que la racine de l'arborescence. A chaque itération, le meilleur nœud est retiré de la liste et est remplacé par tous ses fils sauf s'il s'agit de feuilles. Dans ce cas, la meilleure feuille nouvellement construite est utilisée pour la mise à jour du meilleur chemin actuellement construit. La liste de chemins à étendre est ensuite amputée de tous les nœuds de PED supérieur à un rayon  $r$  fixé au PED de l'actuel meilleur chemin. La détection se termine quand la liste de chemins partiels est vide.

Cette méthode remplit très rapidement la liste puisque pour chaque nœud retiré,  $q$  fils sont ajoutés ce qui requiert une quantité de stockage conséquente. Une amélioration consiste à convertir l'arbre  $q$ -aire en un arbre binaire selon la méthode premier enfant/premier frère [65]. Ainsi, chaque nœud retiré n'est désormais remplacé que par deux nouveaux nœuds (le premier enfant et le premier frère) ce qui limite la mémoire



requis pour stocker la liste. Cette version est appelée parcours par métrique modifié (modified best-first, MBF).

Cependant, quelle que soit la façon de maintenir la liste, ces détecteurs mettent du temps à produire un chemin complet car les chemins retenus pour l'expansion sont souvent proches de la racine. Cette latence porte préjudice aux performances car elle empêche d'amputer la liste en fonction des PED puisqu'il faut un chemin complet pour fixer  $r$ . Pour résoudre ce problème, une variante nommée parcours par métrique modifié avec descente rapide (modified best-first with fast descent, MBF-FD) modifie le traitement effectué lors de l'extension d'un chemin. Plutôt que de seulement ajouter son premier fils et son premier frère, le chemin est étendu jusqu'à atteindre une feuille en ajoutant à la liste tous les frères rencontrés [66]. Ainsi, chaque nœud extrait de la liste conduit à un chemin complet qui permet d'amputer la liste et donc d'accélérer la détection.

Ces principes généraux peuvent être adaptés pour faciliter l'implémentation sur ASIC en divisant la liste selon le niveau dans l'arbre [4]. À chaque itération, le meilleur nœud de chaque liste est extrait, son meilleur frère est ajouté à la liste du même niveau et son meilleur fils est transmis à la liste du niveau inférieur. La détection se termine cette fois quand toutes les listes sont vides. Ces étapes sont reprises graphiquement dans la figure 1.14. Cette variante rend chaque niveau indépendant des autres ce qui permet une très forte parallélisation matérielle. De plus, la taille des listes peut être choisie séparément ce qui offre un contrôle très fin des ressources mémoires nécessaires. Ces astuces permettent à l'implémentation de [4] de dépasser un débit 800 Mb/s dans les mêmes conditions que les précédentes implémentations (64-QAM et canal 4x4).

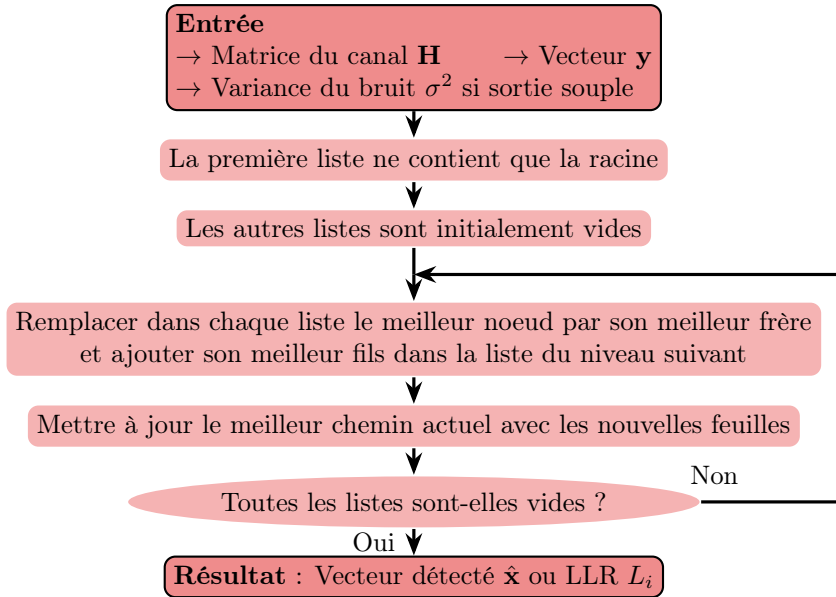


Figure 1.14) Résumé du parcours d'arborescence par métrique tel que décrit dans [4].

### 1.3.5 Apprentissage par réseaux de neurones

La détection MIMO n'a pas été laissée de côté par l'essor des réseaux neuronaux. Deux paradigmes sont à distinguer : l'apprentissage hors-ligne et l'apprentissage en ligne.

L'apprentissage hors-ligne s'intéresse à la recherche d'un réseau de neurones capable de résoudre le problème pour toute matrice de canal après un entraînement préalable. Les données d'entraînement doivent donc contenir une grande variété de conditions afin que le réseau puisse généraliser et traiter une matrice nouvelle sans erreur. Après un apprentissage suffisant, ce type de détecteur peut donc s'attaquer à toutes les réalisations de la matrice de canal pour peu qu'elle suive une dynamique représentée dans l'ensemble d'entraînement. L'un des premiers réseaux de détection performants est DetNet qui s'inspire de la méthode de descente de gradient avec une fonction objectif visant à améliorer les performances vis à vis des détecteurs linéaires [67]. Ce réseau est entraîné pour des canaux de Rayleigh non corrélés.

L'apprentissage hors-ligne s'est aussi étendu à l'apprentissage de bout en bout qui vise à remplacer l'ensemble du récepteur voir l'ensemble de la chaîne de transmission par un ou plusieurs réseaux neuronaux. Un exemple typique est un réseau de neurones entraîné pour réaliser simultanément l'estimation du canal, la détection et la démodulation [68]. Le parti pris est de réaliser chaque fonction dans un réseau de neurones spécialisé au lieu d'avoir un seul réseau pour toute la chaîne. Cette méthode a l'avantage de proposer des performances équivalentes aux autres méthodes d'apprentissage tout en apportant une plus grande flexibilité. En effet, chaque sous-composant peut être modifié sans avoir besoin de ré-entraîner l'ensemble du récepteur.

L'apprentissage hors-ligne pose cependant la question de la stabilité de l'apprentissage. En effet, si le canal réel a une dynamique trop variable ou si celle-ci n'était pas présente dans l'ensemble d'apprentissage, le récepteur n'atteindra pas les performances escomptées. Pour cela, des réseaux plus simples ont été développés dans l'optique d'être moins généralisables mais de pouvoir apprendre assez vite pour s'adapter à la volée [69]. Cette approche permet d'apprendre chaque réalisation de la matrice de canal et donc d'être particulièrement robuste vis à vis de la dynamique du canal.

### 1.3.6 Détecteurs bio-inspirés

Les détecteurs bio-inspirés les plus étudiés dérivent tous des optimisations par évolution de population avec notamment les optimisation par colonnies de fourmis (ant colony optimizations, ACO) et l'FA. Ces détecteurs fonctionnent de façon itérative avec un symbole détecté par itération. Il s'agit toujours de métaheuristiques non déterministes possédant une part de hasard dans leurs exécutions.

#### Optimisation par colonie de fourmis

Les détecteurs exploitant les optimisation par colonnies de fourmis (ant colony optimizations, ACO) simulent un essaim de fourmis qui suivent une trajectoire aléatoire selon une loi non uniforme [70]. A chaque itération, une fourmi sélectionne le symbole

$s \in \mathcal{Q}^{N_t}$  avec la probabilité

$$p_s = \frac{\tau_s^\alpha \eta_s^\beta}{\sum_{s \in \mathcal{Q}^{N_t}} \tau_s^\alpha \eta_s^\beta} \quad (1.26)$$

avec  $\tau_s$  le niveau de phéromones sur le chemin,  $\eta_s$  une image de la fonction objectif (la plupart du temps par une fonction log-sigmoïde) et  $(\alpha, \beta)$  les deux paramètres qui équilibrent l'importance relative de chaque terme. Après chaque itération, le niveau de phéromones est mis à jour selon le principe suivant : plus la fonction objectif de la fourmi est bonne, plus elle laisse de phéromones sur son trajet. Ainsi, les fourmis suivantes sélectionnent plus souvent un chemin prometteur au regard des itérations précédentes tout en conservant une chance d'explorer un nouveau chemin.

### Algorithme des lucioles

La métaheuristique FA est un outil d'optimisation qui simule des lucioles volant vers leur congénères les plus attractifs. Cette attractivité est déterminée en fonction de la valeur de la fonction objectif au point qu'elle occupe. Ainsi, les lucioles les plus attractives sont celles les plus proches de l'optimum recherché. A chaque itération, la luciole  $i$  bouge vers sa congénère  $j$  seulement si cette dernière est plus attractive pour que l'essaim se regroupe au niveau de l'optimum. De plus, le mouvement est superposé à une marche aléatoire pour intégrer un terme d'exploration [71].

La métaheuristique FA a été adaptée à la détection MIMO en utilisant deux types de lucioles : des lucioles fixes et des lucioles mobiles [2]. La figure 1.15 présente l'interprétation de la détection FA dans le cas simplifié de deux antennes utilisant une BPSK. Chaque antenne est représentée par un nid abritant une luciole pour symbole de la constellation. L'attractivité de ces lucioles dépend de leur fonction objectif ainsi que du chemin précédemment parcouru. Ainsi, deux lucioles mobiles n'ayant pas fait le même chemin pourront trouver plus ou moins attractive l'une des lucioles fixes.

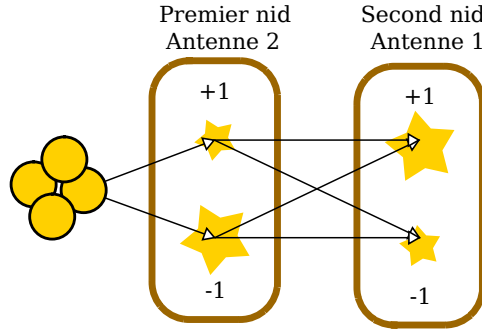


Figure 1.15) Interprétation de la détection FA pour deux antennes et pour une BPSK.

La détection commence avec un essaim de luciole placé en amont des nids. Chaque luciole mobile doit choisir un partenaire dans chaque nid en fonction de l'attractivité additionnée à un bruit gaussien. Une fois toutes les lucioles arrivées dans le dernier nid,

on compare l'attractivité cumulée qu'elles ont rencontrée et la meilleure détermine le vecteur détecté par ses choix. Une nouvelle interprétation de ce détecteur ainsi qu'une généralisation seront proposées dans le chapitre 2 de ce manuscrit.

### 1.3.7 Détecteurs géométriques

Les détecteurs géométriques ont été très peu discutés depuis leur introduction dans la fin des années 2000 [6, 7] mais nous les incluons dans cet état de l'art pour deux raisons. Tout d'abord, ces détecteurs utilisent une technique qui mêle exploration et exploitation d'une façon unique. Ensuite, ce type de détecteur sert de base pour le développement de nouveaux algorithmes qui seront détaillés dans la suite du manuscrit (Cf. chapitres 3 et 4).

La détection géométrique repose sur deux étapes consécutives : une exploration pour trouver un petit ensemble de solutions prometteuses et une exploitation pour améliorer cet ensemble à faible coût. Ce processus peut être rapproché d'une des méthodes classiques en optimisation non convexe : mener plusieurs descentes simples depuis des points de départ différents pour obtenir un maximum local et retenir le meilleur comme optimum global. Ce type de méthode fonctionne bien tant que l'optimum est atteint par au moins une descente ce qui pose la question de la qualité des points de départ.

Les détecteurs géométriques reposent sur la SVD de la matrice du canal  $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$  avec  $\mathbf{U}$  et  $\mathbf{V}$  deux matrices orthogonales et  $\mathbf{\Lambda}$  la matrice diagonale contenant les valeurs singulières  $\{\lambda_i : i \in \llbracket 1, n \rrbracket\}$  en ordre croissant. La fonction objectif peut alors être réécrite comme

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \left(\mathbf{V}^T(\mathbf{x} - \mathbf{x}_0)\right)^T \mathbf{\Lambda} \mathbf{U}^T \mathbf{U} \mathbf{\Lambda} \left(\mathbf{V}^T(\mathbf{x} - \mathbf{x}_0)\right) \quad (1.27)$$

en utilisant  $\mathbf{x}_0$  de (1.11). Comme les vecteurs de  $\mathbf{V}$ , notés  $\{\mathbf{v}_i : 1 \leq i \leq n\}$ , constituent une base, on peut définir  $\{\alpha_i : 1 \leq i \leq n\}$  les coordonnées de  $\mathbf{x} - \mathbf{x}_0$  sur cette base. En utilisant l'orthogonalité de  $\mathbf{U}$  et  $\mathbf{V}$  et la diagonalité de  $\mathbf{\Lambda}$ , cela conduit à

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \sum_{i=1}^n \alpha_i^2 \lambda_i^2. \quad (1.28)$$

Soit  $\Delta_i$  la ligne droite passant par  $\mathbf{x}_0$  et dirigée par  $\mathbf{v}_i$ . On peut noter que (1.28) souligne que la fonction objectif croît plus lentement le long des premiers  $\Delta_i$  plutôt que le long des derniers. Ainsi, l'optimum a plus de chance de se trouver le long des premières lignes associées aux valeurs singulières les plus faibles. L'exploration géométrique revient donc à sélectionner quelques points proches des premiers  $\Delta_i$ . Par la suite, le détecteur effectue des descentes simples depuis chacun de ces points prometteurs. Le détecteur peut également produire des sorties souples en exploitant l'approximation max-log et une liste telle que présentée à l'équation (1.25) [6].

## 1.4 Synthèse de l'état de l'art et problématique des travaux

Cette section commence par la synthèse de l'état de l'art qui vient d'être présentée afin de mettre en lumière certaines similitudes et différences entre les paradigmes. Ces caractéristiques permettent ensuite de proposer la problématique qui dirige l'ensemble des travaux décrits dans ce manuscrit.

### 1.4.1 Synthèse des stratégies de détection

Le tableau 1.2 propose une synthèse des stratégies de détection et des références de chaque détecteur introduit précédemment ainsi que certains critères qui nous seront utiles pour établir la problématique de ce manuscrit :

**L'optimalité** différencie les détecteurs optimaux à complexité exponentielle des détecteurs polynomiaux sous-optimaux. La variante sous-optimale du détecteur SD utilisant une énumération SE est omise pour permettre une lecture plus facile du tableau.

**L'exploitation de modèle** indique si le détecteur exploite certaines caractéristiques particulières du problème de détection (1.10) ou s'il applique une métaheuristique générale.

**L'exploration** liste les détecteurs qui réalisent, à une étape donnée, des choix sous-optimaux dans l'espoir d'obtenir un meilleur résultat final. Cela s'oppose à la stratégie qui prend le meilleur choix à chaque étape. Nous utilisons ce terme par analogie au dilemme exploration-exploitation présent dans le milieu de l'apprentissage par renforcement. Ce dernier affirme qu'un algorithme optimal doit nécessairement mélanger ces deux comportements sous peine d'obtenir de très mauvais résultats [72].

Noms et acronymes	Principe	Exploration	Exploitation de modèle	Optimalité	Références
<b>IC</b>	Hypothèses & annulation	Non	Oui	Non	[45, 47, 49, 50, 53]
<b>DFS : SD</b>	Recherche dans une sphère	Non	Oui	Oui	[43, 55, 58, 59]
<b>BFS : K-best</b>	Conservation de $K$ chemins par niveau	Non	Oui	Non	[3, 60–63]
<b>MFS</b>	Priorisation des nœuds prometteurs.	Non	Oui	Non	[4, 66]
<b>Bio-inspirés</b>	Métaheuristique d'optimisation	Oui	Non	Non	[2, 70, 71]
<b>Géométrique</b>	Exploration basée sur la SVD et optimisation locale	Oui	Oui	Non	[6, 7]
<b>Réseau de neurones</b>	Apprentissage puis détection	Non	Non	Non	[67–69]

Tableau 1.2) Comparaison des détecteurs présentés selon différents critères d'intérêt.

### 1.4.2 Différence entre le problème de détection théorique et celui résolu en pratique

Tous les détecteurs recensés dans notre état de l'art cherchent à résoudre le problème de détection (1.10). Comme l'introduction de la section 1.3 l'indiquait déjà, minimiser la fonction objectif revient à minimiser l'amplitude du bruit. En effet, le modèle (1.9) donne directement  $\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{w}\|^2$ . Or le bruit  $\mathbf{w}$  est supposé suivre une loi normale centrée, ce qui implique que les bruits de faible puissance sont les plus probables. Résoudre (1.10) est donc la façon optimale de rechercher les symboles les plus probablement émis. C'est l'approche qui a motivé le développement de la quasi-totalité des détecteurs précédents.

Cependant, le problème de détection tel qu'il est réalisé en pratique est plus difficile que sa formulation théorique. En effet, le problème de détection théorique suppose que la seule source d'erreur est le bruit gaussien issu du canal de transmission. Or, le problème de détection à résoudre en pratique possède d'autres sources d'erreurs qui ne sont pas prises en compte dans la formulation de (1.10). Nous listons ici trois exemples de telles sources d'erreur :

#### Détection à sorties souples et contre-hypothèses

Tout d'abord, le codage de canal est un incontournable de toutes les transmissions modernes. Ainsi, les détecteurs modernes doivent en réalité proposer des sorties souples, *ie* des LLR. L'approximation max-log avec liste (1.25) transforme le problème en la recherche de deux minima par bit : l'optimum global et la meilleure contre-hypothèse. Or, les détecteurs recensés dans l'état de l'art ne se concentrent que sur la constitution de la liste  $\mathcal{S}$  des meilleurs candidats sans précaution particulière sur le fait qu'ils appartiennent à l'un ou l'autre des  $\chi_i^k$ . Ainsi, ils n'apportent aucune garantie sur la présence de la contre-hypothèse dans  $\mathcal{S}$  ce qui introduit une source d'erreur qui n'existait pas dans le problème théorique initial.

#### Approximation algorithmique pour maîtriser la complexité

Ensuite, tous les détecteurs utilisent une résolution sous-optimale afin de limiter la complexité algorithmique. La synthèse du tableau 1.2 le montre très bien puisque seul le détecteur SD sans énumération SE est mathématiquement optimal. Si l'intérêt de ces simplifications algorithmiques est évident, il introduit forcément des erreurs qui ne sont pas prises en compte par le modèle théorique. Par exemple, en ne conservant que  $K$  nœuds par niveau, le détecteur K-best élague des chemins qui auraient pu mener à des meilleures solutions. Si l'on assimile ces erreurs d'approximation à un nouveau bruit de calcul, alors nous ne connaissons rien de sa distribution ou de ses caractéristiques. Ainsi, la minimisation de  $\|\mathbf{w}\|^2$  n'est plus nécessairement équivalente à la recherche du meilleur vecteur émis.

### Quantification pour l'implémentation matérielle

Enfin, ces algorithmes sont destinés à être implémentés sur des architectures matérielles possédant leurs propres contraintes. L'une des contraintes majeures est la quantification des variables sur un nombre réduit de bits, le plus souvent avec un codage en virgule fixe. La littérature rapporte des quantifications sur une dizaine de bits [62, 71, 73] ce qui implique un niveau de bruit non négligeable. Encore une fois, ce bruit de quantification n'est pas pris en compte dans la formulation du problème de détection ce qui peut créer un écart entre la solution optimale pour des variables quantifiées et celle pour des variables réelles.

#### 1.4.3 Problématique et structure du manuscrit : de l'intérêt de l'exploration pour la détection MIMO

La résolution optimale du problème de détection dans sa formulation théorique est logiquement abordée par des algorithmes déterministes connus pour leur performances : parcours par *branch and bound*, algorithme de Dijkstra, *etc.* Cependant, nous avons souligné que d'autres sources d'erreurs s'ajoutaient au bruit gaussien et que ces sources n'étaient pas prises en compte dans la formulation théorique. De plus, ces sources ne sont pas évidentes à modéliser, d'autant plus qu'elles peuvent interagir les unes avec les autres. Enfin, d'autres sources supplémentaires pourraient entrer en ligne de compte comme une mauvaise estimation du canal ou un modèle de canal trop simpliste (Cf. section 1.2.2).

Toutes ces sources d'erreurs supplémentaires font que le choix d'un algorithme adapté pour le problème théorique n'est pas forcément idéal pour résoudre le problème en pratique qui présente des incertitudes supplémentaires. Or, les métaheuristiques bio-inspirées sont très performantes dans des conditions dégradées mettant en défaut les autres classes de détecteurs. Un détecteur par ACO peut par exemple approcher des performances optimales dans le cas d'un canal corrélé [70] réputé beaucoup plus complexe.

L'hypothèse de ces travaux de thèse est que la résilience des détecteurs bio-inspirés provient de leur propension à explorer l'espace des possibles plutôt que de prendre le meilleur choix à chaque étape. Cependant, se priver totalement des connaissances disponibles pour se réduire à appliquer une métaheuristique générale revient à se priver d'informations utiles. Ainsi, ces travaux de thèse poursuivent un double objectif : la validation de l'hypothèse précédente et le développement d'un détecteur combinant les connaissances issues du modèle et l'exploration.



## 2 Intérêt de l'exploration

### Développement et étude d'un détecteur à parcours d'arborescence stochastique

Le chapitre précédent a d'abord introduit une modélisation du canal et de la transmission ainsi qu'une modélisation du problème de détection en lui-même. Ainsi, la détection a, par exemple, été présentée comme un parcours d'arborescence ou une annulation d'interférences mutuelles. Ce chapitre fait suite à un double constat issu l'état de l'art qui a suivi et nous rappelons rapidement ici ce constat directeur.

Tout d'abord, le modèle offre des connaissances exploitables comme le prouvent les détecteurs linéaires, par annulation des interférences (interferences cancellation, IC) ou par parcours d'arborescence. Le modèle permet de traduire et de simplifier la situation réelle en une structure mathématique formelle. Il devient alors possible d'utiliser des algorithmes bien connus pour résoudre efficacement le problème issu de la modélisation.

D'autre part, les métaheuristiques génériques telles que les optimisations bio-inspirées atteignent des performances intéressantes avec peu ou pas de connaissances du modèle. En effet, elles utilisent quasiment les mêmes outils pour le problème de détection 1.10 que pour les autres problèmes d'optimisation. De plus, ces algorithmes se montrent plus résilients et nous supposons que l'exploration joue un rôle dans cette caractéristique.

C'est pourquoi nous nous demandons si *les détecteurs exploitant le modèle deviennent plus résilients après ajout d'un comportement exploratoire*. Ce chapitre propose une évaluation de cette proposition par la comparaison de stratégies ne différant que par leur capacité à explorer ou non. Pour ce faire, la section 2.1 présente les métriques qui permettent de comparer deux détecteurs entre eux et la section 2.2 définit le vocabulaire des parcours d'arborescence. Par la suite, la section 2.3 développe un nouveau détecteur à parcours d'arborescence stochastique similaire aux parcours d'arborescence de la littérature mais ajoutant une exploration aléatoire. La section 2.4 détaille les deux références à parcours d'arborescence qui serviront de référence pour les analyses de la section 2.5. Enfin, ce chapitre conclut sur l'hypothèse dans la section 2.6 en montrant que l'ajout d'exploration a bien amélioré les performances du détecteur.

### 2.1 Méthodes et critères d'évaluation

Il est nécessaire de définir les critères d'évaluation des détecteurs puisque la validation de l'hypothèse s'appuie sur la comparaison entre un nouveau détecteur et des stratégies de l'état de l'art. Nous retenons les deux critères les plus communs dans la littérature : les performances en terme de taux d'erreurs binaires (bit error ratio, BER) en

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

fonction du rapport signal à bruit (signal-to-noise ratio, SNR) et la complexité. Les sections 2.1.1 et 2.1.2 décrivent respectivement l'évaluation des taux d'erreur et de la complexité et la section 2.1.3 introduit l'analyse de Pareto qui permet de coupler les deux métriques précédentes.

### 2.1.1 BER en fonction du SNR : la méthode de Monte Carlo

La métrique la plus évidente pour comparer des détecteurs est de tracer le BER en fonction du SNR. Le meilleur détecteur étant évidemment celui qui commet le moins d'erreurs à un SNR donné ou réciproquement qui requiert le SNR le plus faible pour atteindre un BER fixé.

Le SNR est défini par

$$\text{SNR} = \frac{\sum_{i=1}^{N_t} \mathbb{E}_{\mathcal{Q}_k}(x_k^2)}{rN_0} \quad (2.1)$$

avec  $\mathbb{E}_{\mathcal{Q}_k}(x_k^2)$  la puissance moyenne du  $k^{\text{ème}}$  flux de donnée définie par (1.15),  $N_0$  la variance du bruit,  $r$  le rendement du code de canal s'il est présent et  $N_t$  le nombre d'antennes d'émission. Sans code de canal, le rendement est fixé à 1. Dans le cas classique où toutes les antennes utilisent la même constellation à la même puissance, le SNR se simplifie en

$$\text{SNR} = \frac{N_t E_Q}{rN_0} \quad (2.2)$$

avec  $E_Q$  la puissance moyenne de la constellation.

Obtenir l'expression exacte de cette fonction par des développements mathématiques est bien souvent trop ardu pour être réalisable. C'est pourquoi la littérature a recours à la méthode de Monte Carlo pour estimer le taux d'erreur à SNR fixé.

Cette méthode considère le BER comme une variable aléatoire qui prend une valeur différente à chaque tirage en fonction du message binaire et de l'état du canal. On note  $B$  la variable aléatoire ainsi formée. Il est alors possible de tirer  $t$  fois  $B$  pour estimer la valeur du BER. Pour cela, on introduit la nouvelle variable aléatoire

$$\hat{B}_t \triangleq \frac{1}{t} \sum_{\theta=1}^T B_\theta \quad (2.3)$$

représentant la moyenne empirique des  $t$  tirages. Le théorème central limite implique que  $\hat{B}_t$  converge en distribution vers une loi normale :

$$\hat{B}_t \xrightarrow[t \rightarrow \infty]{} \mathcal{N}\left(\mathbb{E}(B), \frac{\sigma_B}{n}\right) \quad (2.4)$$

avec  $\mathbb{E}(B)$  le BER recherché et  $\sigma_B$  l'écart-type du BER entre les différentes réalisations.

Ainsi, le BER recherché peut être estimé par la moyenne empirique des tirages. De plus, l'équation (2.4) fournit aussi l'écart-type  $\sigma_{\hat{B}}$  de l'estimation

$$\sigma_{\hat{B}} = \frac{\sigma_B}{\sqrt{t}}. \quad (2.5)$$

qui montre qu'il est donc possible d'avoir une estimation aussi fine que voulu du BER à condition de faire suffisamment de tirages. Cependant, comme l'écart-type de l'estimation décroît avec la racine de  $t$ , diviser l'incertitude par 10 requiert 100 fois plus de tirages.

### 2.1.2 Évaluation de la complexité arithmétique

La deuxième métrique de comparaison usuelle est la complexité algorithmique qui comprend plusieurs déclinaisons : complexité temporelle (temps de calcul), complexité spatiale (espace mémoire nécessaire) et complexité arithmétique (nombre d'opérations). Toutes ces complexités varient logiquement avec la taille des entrées, dans notre cas avec le nombre d'antennes du système. Dans le cas d'algorithmes dont la complexité varie entre deux entrées de même taille, nous utiliserons toujours les complexités moyennes estimées par la méthode de Monte Carlo décrite précédemment.

#### Influence de la cible matérielle sur le choix de la métrique de complexité

Nous faisons le choix de nous concentrer uniquement sur la complexité arithmétique par vecteurs détectés en discriminant les types d'opérations. Ce choix s'explique par la volonté d'implémenter les algorithmes sur des FPGA et des ASIC. En effet, sur ces plateformes, une addition, une multiplication, une comparaison et un changement de signe ne requièrent pas du tout les mêmes ressources, ce qui n'est pas pris en compte dans la complexité temporelle dans son sens classique. Ainsi, la complexité arithmétique est une métrique plus appropriée pour estimer le débit ou la consommation énergétique sur ces plateformes.

Sur un ASIC, nous considérons que le produit est l'opération la plus coûteuse, suivie par l'addition. La comparaison et le changement de signe sont considérés négligeables devant les autres opérations. La hiérarchie reste la même sur un FPGA classique composé uniquement de blocs logiques.

Cependant, les FPGA plus récents embarquent également des blocs optimisés tels que des processeurs de signaux numériques (digital signal processors, DSP) et des blocs de mémoire vive (random-access memory, RAM). La présence de ces DSP optimise les produits-accumulations ce qui implique qu'un produit a exactement le même coût qu'un produit-accumulation. Dès lors, les additions deviennent négligeables si elles peuvent être adjointes à un produit, ce qui est quasiment toujours possible puisque les calculs prépondérants sont des multiplications matricielles et des calculs de normes.

#### Études de complexité asymptotique

La complexité est souvent étudiée en fonction de la taille de l'entrée (*ie*, le nombre d'antennes du système) à travers des comparaisons asymptotiques. La taille de la constellation est également un facteur d'influence mais elle ne tend pas vers l'infini et n'entre pas en ligne de compte dans l'étude asymptotique. Le but est alors de comparer la vitesse de croissance d'une fonction (ici la complexité) avec des fonctions connues telles que des polynômes ou des exponentielles. Pour ces comparaisons, nous

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

adoptons les notations de Landau usuelles que nous définissons ici pour les suites à valeurs positives [65].

- $f(n)$  *major*e à une constante près ou *domine*  $g(n)$  s'il existe  $k > 0$  tel que  $f(n) \leq kg(n)$  à partir d'un certain rang. On note alors  $f(n) = O(g(n))$ .
- $f(n)$  *est minorée* à une constante près par ou *est soumise* à  $g(n)$  s'il existe  $k > 0$  tel que  $kg(n) \leq f(n)$  à partir d'un certain rang. On note alors  $f(n) = \omega(g(n))$ .
- $f(n)$  *est de l'ordre de*  $g(n)$  s'il existe deux coefficients  $k_1 > 0$  et  $k_2 > 0$  tels que  $k_1g(n) \leq f(n) \leq k_2g(n)$  à partir d'un certain rang. On note alors  $f(n) = \Theta(g(n))$ .

### 2.1.3 Analyse de Pareto

Nous venons de définir les deux axes considérés dans notre démarche d'amélioration des détecteurs MIMO. Ainsi, nous recherchons des algorithmes commettant peu d'erreurs et ayant une complexité arithmétique faible. Cependant, l'état de l'art montre que ces deux critères sont antagonistes puisque les approximations réduisant la complexité engendrent des erreurs. L'analyse de Pareto est alors un bon outil pour étudier le compromis entre ces deux critères dichotomiques.

La figure 2.1 illustre le besoin d'un outil objectif d'analyse des critères antagonistes en comparant deux détecteurs à sortie souple : le K-best de [3] avec  $K = 128$  et le best-first parallèle de [4] avec des listes de tailles 1, 3 et 5. Les métriques sont obtenues pour un canal 4x4, une 16-QAM et un code LDPC. Sur la figure 2.1(a), le best-first se trouve distancé en requérant environ 2 dB de plus que le K-best pour obtenir un taux d'erreur de  $10^{-4}$ . Cependant, la comparaison de complexité arithmétique de la figure 2.1(b) montre que ce même K-best requiert quasiment 10 fois plus de produits que son concurrent. Ainsi, en ne se concentrant que sur l'une ou l'autre des métriques, il est possible de conclure à la supériorité de l'un ou l'autre des détecteurs. Pour autant, en regardant les deux métriques simultanément, ces deux détecteurs sont équitablement pertinents puisqu'ils règlent simplement le curseur du compromis performance/complexité différemment.

L'analyse de Pareto permet une analyse conjointe des deux critères étudiés. Pour cela, on dit que le détecteur  $D_1$  *domine* le détecteur  $D_2$  si  $D_1$  est meilleur pour un critère et au moins équivalent pour l'autre. On note alors  $D_2 \prec D_1$  ce qui permet de former un ordre partiel sur les détecteurs. En effet, l'ordre n'est pas total car il existe des couples de détecteurs pour lesquels aucun ne domine l'autre. C'est par exemple le cas de l'exemple précédent.

Lors du choix d'un détecteur, il n'y a alors aucun intérêt à choisir un détecteur dominé puisque prendre son dominant offre des meilleures performances pour au moins un critère sans perdre sur l'autre. Les meilleurs algorithmes sont donc ceux qui ne sont pas dominés. Cet ensemble est nommé front de Pareto et il peut être défini formellement par

$$\mathcal{P} \triangleq \{D_1 : \nexists D_2, D_1 \prec D_2\}. \quad (2.6)$$

L'ensemble des détecteurs du front de Pareto sont dits *efficaces* ou *optimaux* au sens de Pareto.

La figure 2.2 représente ce principe dans les mêmes conditions que précédemment avec un best-first et plusieurs K-best réglés différemment. On retrouve tout d'abord l'idée que le best-first et le K-best avec  $K = 128$  sont tous les deux efficaces et qu'aucun ne domine l'autre. En effet, passer de l'un à l'autre détériore au moins un critère. Ainsi, choisir entre l'un et l'autre ne relève que d'un choix dans le compromis performance/-complexité. Par contre, le K-best avec  $K = 32$  n'appartient pas au front car il est par exemple dominé par le best-first. Il n'y a donc aucun intérêt de choisir ce détecteur.

L'analyse de Pareto permet donc une comparaison objective et conjointe des deux critères et permet une représentation graphique efficace. Par la suite, nous utiliserons cet outil pour établir si un détecteur a été amélioré. Ainsi, un détecteur est supérieur à l'état de l'art dès lors qu'il est efficace au sens de Pareto. De plus, un détecteur est flexible s'il est efficace au sens de Pareto pour beaucoup de points de fonctionnement différents.

## 2.2 Vocabulaire du parcours d'arborescence

Cette section définit l'ensemble des termes relatif au parcours d'arborescence qui seront utilisés par la suite. Pour ce faire, un exemple de d'arborescence est proposé sur la figure 2.3.

Les *nœuds* sont les éléments de base de l'arborescence et ils sont liés entre eux par des *arêtes* orientées. Ici, les nœuds sont représentés par des numéros de 1 à 4, des lettres *a* à *d* ou par le mot *Racine*. La position relative des nœuds est exprimée par un vocabulaire généalogique. Ainsi, le *père* d'un nœud est le nœud qui le précède (la racine est le père de 3) et la notion de *fils* est définie par analogie (3 est le fils de la racine). Enfin, deux nœuds ayant le même père sont dit *frères* (2 et 3 sont frères).

Une arborescence contient deux type de nœuds particuliers : les feuilles et la racine. Une *feuille* est un nœud qui ne possède aucun fils (ici les lettres). La *racine* est, quant à elle, le seul nœud qui ne possède pas de père. Dans une arborescence, tous les nœuds peuvent être atteints par une relation de filiation depuis la racine.

L'ensemble des nœuds et des arêtes forment l'arborescence. On parle d'arborescence *q*-aire quand chaque nœud possède au plus *q* fils. Notre exemple présente donc une arborescence 4-aire. Enfin, on appelle *niveau* de l'arborescence tous les nœuds qui sont à la même distance de la racine et on nomme *profondeur* le plus grand niveau. L'arborescence de la figure 2.3 possède ainsi une profondeur de 2, le premier niveau étant formé des nœuds numérotés et le second des nœuds nommés par des lettres. Enfin, l'arborescence se différencie de l'arbre par l'unicité de sa racine et par le fait que tous les nœuds ont un unique parent.

Dans le cas du problème de détection, les arborescences utilisées sont *pondérées*, c'est à dire que chaque arête porte un poids. D'après la section 1.3.4, les poids sont fixés en fonction de (1.22). On nomme distance euclidienne partielle (partial Euclidian distance, PED) le poids cumulé d'un nœud, c'est à dire la somme de tous les poids rencontrés sur les arêtes depuis la racine jusqu'à ce nœud. Le problème de détection vise alors à trouver le chemin menant à la feuille de poids cumulé le plus faible puisque le PED d'une feuille correspond à la fonction objectif d'après (1.20).

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

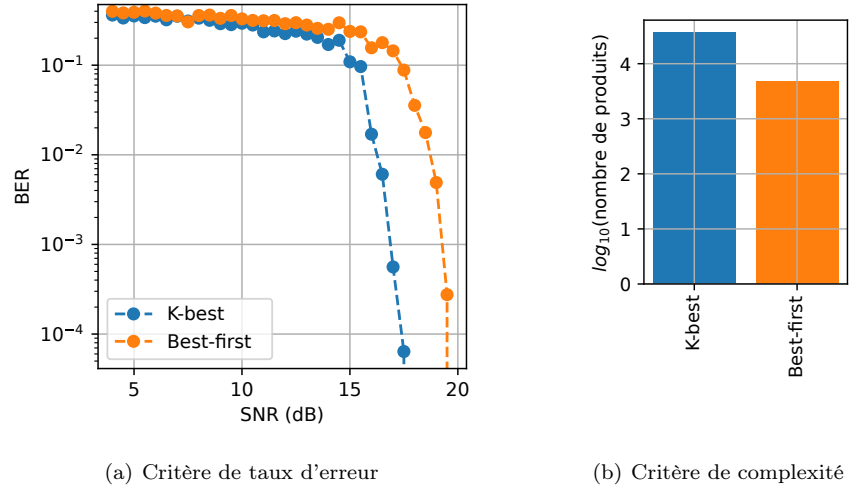


Figure 2.1) Exemple de comparaison de performances entre un K-best de [3] avec  $K = 128$  et un best-first parallèle de [4] avec des listes de tailles (1, 3, 5) pour un canal 4x4, une 16-QAM et un code LDPC.

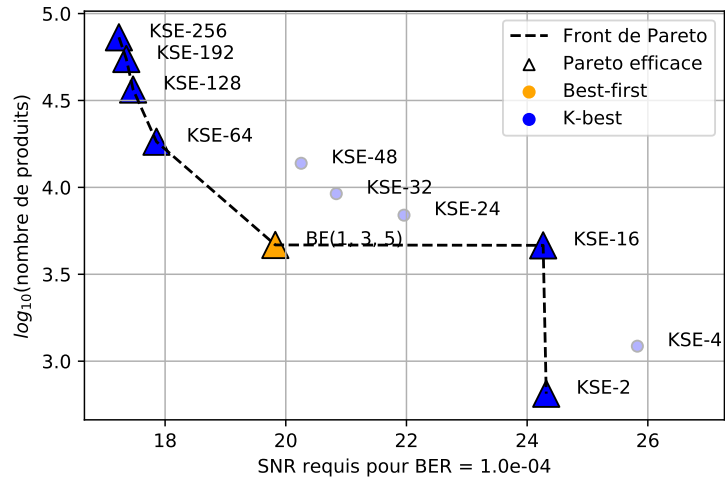


Figure 2.2) Analyse de Pareto des deux critères antagonistes.

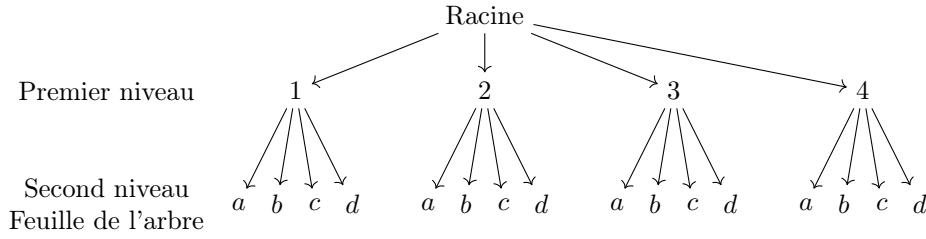


Figure 2.3) Exemple d'une arborescence à deux niveaux avec quatre fils par nœud.

## 2.3 Un nouveau détecteur à parcours d'arborescence stochastique

Après la définition des critères et d'un ordre partiel permettant une comparaison objective des algorithmes, il nous faut maintenant proposer un détecteur ne différant de l'état de l'art que par son comportement exploratoire. Cette section développe ce nouveau détecteur en ajoutant aux parcours d'arborescence une composante d'exploration inspirée des métaheuristiques bio-inspirées. La section 2.3.1 introduit l'inspiration bio-inspirée et la rapproche des parcours d'arborescence classiques. Ensuite, la section 2.3.2 étend le nouveau détecteur à d'autres modulations et aux sorties souples. Enfin, la section 2.3.3 synthétise le fonctionnement de ce nouveau détecteur et en analyse la complexité arithmétique.

### 2.3.1 De la métaheuristique bio-inspirée au parcours d'arborescence stochastique

Le détecteur par algorithme des lucioles (firefly algorithm, FA) est l'application de la métaheuristique correspondante [71] au problème de détection (1.10). La section 1.3.6 de l'état de l'art a déjà introduit rapidement l'algorithme d'optimisation générique et nous nous concentrons maintenant sur le fonctionnement du détecteur en lui-même selon la version de [2]. Cette dernière publication présente un détecteur restreint à la résolution du problème à sortie dure avec une modulation par changement de phase en quadrature (quadrature phase-shift keying, QPSK).

Dans le paradigme bio-inspiré, une antenne de réception correspond à un nid contenant une luciole stationnaire par symbole de la constellation. Le détecteur simule un essaim de  $F$  lucioles mobiles qui doivent choisir un partenaire par nid en fonction de leur attractivité. Les  $F$  insectes partent simultanément et indépendamment depuis le nid correspondant à la dernière des  $N_t$  antennes et remontent tous les nids jusqu'au premier. Une fois tous les trajets complétés, la luciole ayant l'attractivité cumulée la plus forte est retenue pour la construction du vecteur détecté  $\hat{\mathbf{x}}$ . La figure 2.4(a) illustre cette idée en représentant par des disques les lucioles mobiles et par des étoiles celles qui restent stationnaires. La taille de l'étoile est une image de l'attractivité de cette luciole. Ainsi, une fois que l'essaim a terminé son parcours, si au moins une

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

luciole a sélectionné le partenaire le plus attractif dans chaque nid, le vecteur détecté sera  $\hat{\mathbf{x}} = (+1, -1)$ .

L'attractivité perçue par les lucioles dépend de ses choix précédents et elle est évaluée en utilisant la décomposition QR de la section 1.3.4. Ainsi, l'attractivité de la luciole stationnaire  $s$  par rapport à la luciole mobile  $m$  est définie par

$$\beta_{s,m} \triangleq e^{-\gamma d_{s,m}^k} \quad (2.7)$$

avec  $(\gamma, k) \in \mathbb{R}_{+*}^2$  deux paramètres de réglage et  $d_{s,m}$  la distance calculée en utilisant (1.22). Comme attendu, une luciole est d'autant plus attractive que l'incrément de PED  $d_{s,m}^k$  est faible. L'attractivité va donc dans le sens de la minimisation de la fonction objectif puisqu'une luciole est d'autant plus attractive qu'elle entraînera une augmentation faible de cette fonction.

Les paramètres  $\gamma$  et  $k$  accentuent ou diminuent les différences d'attractivité entre lucioles stationnaires. Or, une luciole mobile choisit son compagnon en fonction d'une variable aléatoire dont la loi de probabilité dépend des attractivités de toutes les lucioles stationnaires dans le nid. Ainsi, la luciole en mouvement  $m$  choisira le partenaire  $s$  avec la probabilité

$$P_{s,m} = \frac{\beta_{s,m}}{\sum_j \beta_{j,m}} \quad (2.8)$$

où  $\sum_j \beta_{j,m}$  est la somme de l'attractivité de toutes les lucioles dans le nid d'intérêt. Par conséquent, les paramètres  $\gamma$  et  $k$  règlent le compromis exploration-exploitation. En effet, l'équation (2.8) montre que si les attractivités sont très différentes (*ie*,  $\gamma$  et  $k$  grands), alors le choix se porte sur la meilleure luciole avec une probabilité très forte. L'algorithme se concentre donc sur le meilleur choix à chaque étape ce qui caractérise l'exploitation. A l'inverse, si les attractivités sont plus proches (*ie*,  $\gamma$  et  $k$  faibles), tous les choix sont possibles menant à une exploration plus importante.

Il est important de noter que l'attractivité dépend des choix précédents de la luciole mobile via la valeur de  $d_{s,m}$ . Pour autant, cette métrique peut être évaluée en ne connaissant que les choix dans les nids précédents sans aucune hypothèse sur les choix suivants, et ce grâce à la décomposition QR (Cf. section 1.3.4). La figure 2.4(b) représente l'attractivité perçue par la luciole selon qu'elle a choisi l'un ou l'autre des partenaires dans le premier nid. Cette figure commence également à faire apparaître la proximité de cet algorithme avec les parcours d'arborescence. Un trajet de luciole peut ainsi être interprété comme la construction d'un chemin dans une arborescence. Un nœud au  $i^{\text{ème}}$  niveau de l'arborescence correspond alors à un choix dans  $i^{\text{ème}}$  nid.

Interprétée comme un parcours d'arborescence, la détection par FA aboutit à  $F$  chemins indépendants partant de la racine et rejoignant une feuille. Pour cela, FA prolonge les chemins partiels selon un tirage aléatoire biaisé en faveur des nœuds prometteurs. Ainsi, ce type de détecteur peut être vu comme un parcours d'arborescence avec une exploration aléatoire puisqu'il peut prendre des choix localement sous-optimaux à certains nœuds en espérant compenser plus loin le surcoût de fonction objectif.



### 2.3 Un nouveau détecteur à parcours d'arborescence stochastique

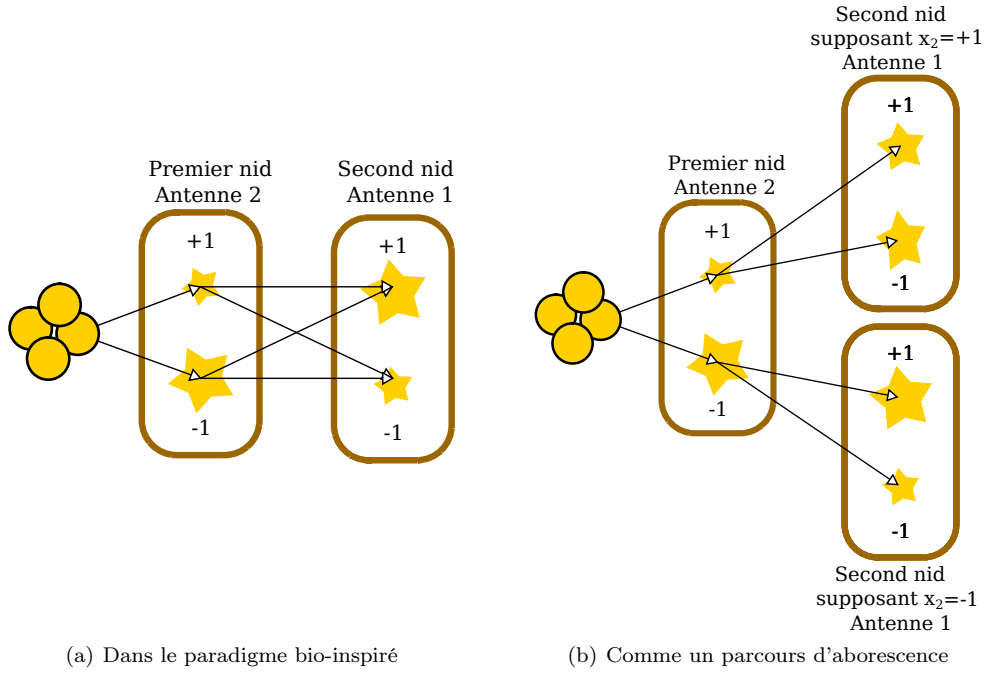


Figure 2.4) Les deux interprétations du détecteur FA dans le cas d'un système 2x2 utilisant une BPSK. Les disques représentent les lucioles mobiles et les étoiles les insectes stationnaires plus ou moins attractifs.

#### 2.3.2 Proposition d'un nouveau détecteur basé sur le FA

Dans cette section, nous proposons un nouveau détecteur par FA construit en améliorant et généralisant le détecteur de [2].

La réinterprétation du détecteur par FA comme un parcours d'arbre stochastique en fait un bon candidat pour évaluer l'intérêt de l'exploration. Cependant, la version de [2] n'est fonctionnelle que pour la détection de QPSK en sortie dure. Cette limite restreint beaucoup l'intérêt de la comparaison puisque les systèmes réels requièrent des sorties souples pour des modulations diverses.

La version décrite précédemment est déjà étendue par nos soins aux modulations d'ordre supérieur, y compris aux modulations qui ne sont pas des QAM. En effet, la probabilité (2.8) est tout à fait compatible avec une modulation comprenant plus de deux symboles et elle ne requiert aucune hypothèse sur la position des symboles dans la constellation. Cette section présente une extension du détecteur de [2] à la production de sorties souples ainsi que des solutions pour implémenter les nouvelles opérations coûteuses que l'exploration par FA implique : le calcul d'exponentielle et la génération de nombres pseudo-aléatoires.

### Généralisation du détecteur aux sorties souples

La proximité entre le détecteur par FA et les détecteurs par parcours d'arborescence permet de s'inspirer de ces derniers pour produire des sorties souples. La section 1.3.4 de l'état de l'art relevait que les algorithmes par parcours d'arborescence estiment les LLR à partir d'une liste  $\mathcal{S}$  et de l'approximation max-log (1.25). Cette méthode se transpose aisément à notre détecteur puisqu'il construit déjà  $F$  chemins par détection. Ainsi, nous construirons dans la suite les LLR à partir de (1.25) et des  $F$  chemins construits.

### Réduction de la complexité par un approximant de Padé

Calculer l'attractivité nécessite d'évaluer une fonction exponentielle, ce qui peut représenter un coût élevé sur les FPGA et ASIC. Des implémentations récentes de calcul numérique par rotation de coordonnées (coordinate rotation digital computer, CORDIC) rapportent des simplifications significatives de l'évaluation de cette fonction [74]. Pour autant, elle reste pénalisante comparée aux autres opérations et une approximation est donc avantageuse.

L'approximation de (2.7) doit en conserver les caractéristiques principales :  $\beta_{s,m}$  est strictement positif et strictement décroissant avec  $d_{s,m}$  sur toutes la plage des  $-\gamma d_{s,m}^k$  susceptibles d'être rencontrés. Une pré-étude montre que cette grandeur varie classiquement entre -2 et 0 et nous retenons donc cet intervalle pour évaluer la qualité de l'approximation. Le développement limité d'ordre  $n$  de l'exponentielle

$$e^x \simeq \sum_{i=0}^n \frac{x^i}{i!} \quad (2.9)$$

ne convient donc pas car il n'est pas décroissant pour  $n$  pair et pas positif pour  $n$  impair. La figure 2.5(a) met en évidence ces défauts sur la plage d'arguments retenue.

Ainsi, nous proposons d'utiliser un approximant de Padé, c'est à dire une approximation par une fraction rationnelle. L'approximant le plus simple répondant aux deux contraintes précédentes est celui d'ordre (0,1) qui est donné par

$$e^x \simeq \frac{1}{1-x} \quad (2.10)$$

ce qui ne requiert qu'une soustraction et une inversion. La figure 2.5(b) met en évidence la qualité de l'approximation ainsi que la validité des deux contraintes recherchées. L'expression (2.7) devient alors

$$\beta_{s,m} \simeq \frac{1}{1 + \gamma d_{s,m}^k}. \quad (2.11)$$

Dans la suite, la complexité et le taux d'erreur seront évalués en utilisant cette approximation.

### 2.3 Un nouveau détecteur à parcours d'arborescence stochastique

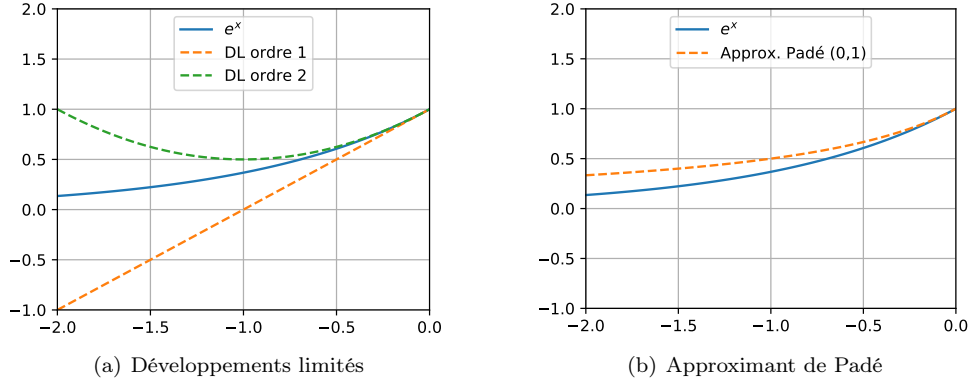


Figure 2.5) Approximations de la fonction exponentielle.

#### Génération de nombres pseudo-aléatoires sur FPGA et ASIC

Tout comme le calcul de l'exponentielle, la génération de nombres pseudo-aléatoires est une nouvelle opération qu'il n'est pas trivial à implémenter. En effet, une détection stochastique doit avoir un comportement similaire à celui qu'elle aurait avec de vrais nombres aléatoires. Il est par exemple nécessaire que les choix aléatoires de fils ne soient pas corrélés les uns aux autres pour obtenir une exploration effective. Pour autant, il n'existe aucune contrainte sur les séquences pseudo-aléatoires entre deux détections distinctes. En effet, le processus de détection reprend de zéro à chaque nouveau vecteur reçu et le cheminement de la précédente détection est totalement indépendant de la suivante. Nous proposons ici deux solutions permettant d'obtenir efficacement une telle séquence pseudo-aléatoire sans aucun produit.

Une première solution est de stocker une séquence suffisamment longue dans un bloc de mémoire morte (read-only memory, ROM) embarqué. En effet, nous venons de dire que la séquence pouvait être identique pour chaque détection tant qu'elle était similaire à une séquence aléatoire au sein d'une même détection. Une détection requiert le tirage de  $FN_t$  nombres différents ce qui représente un millier de valeurs dans les pires cas usuels. Le stockage d'un tel nombre de valeurs est à la portée de la quasi-totalité des plateformes matérielles modernes.

Le stockage en ROM est la solution la plus efficace tant que  $FN_t$  reste relativement faible et que suffisamment de mémoire est disponible. Cependant, il peut être nécessaire de produire les nombres pseudo-aléatoires à la volée dans les rares cas où le nombre d'antennes  $N_t$  ou le nombre de lucioles/chemins  $F$  devient trop important, ou bien dans ceux où aucune mémoire supplémentaire n'est disponible. Pour cela, il est par exemple possible d'utiliser des automates cellulaires auto-programmables (self-programmable cellular automata, SPCA) qui s'implémentent très efficacement sur les plateformes matérielles ciblées [75].

Un automate cellulaire (cellular automaton, CA) unidimensionnel est une chaîne d'éléments logiques appelés cellules. Chaque cellule possède deux entrées et stocke

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

un bit d'état  $E$ . Ces cellules sont ensuite connectées à leur voisine pour former une chaîne. Ainsi les deux entrées de la  $i^{\text{ème}}$  cellule sont les états des cellules  $i + 1$  et  $i - 1$  comme le montre la figure 2.6.

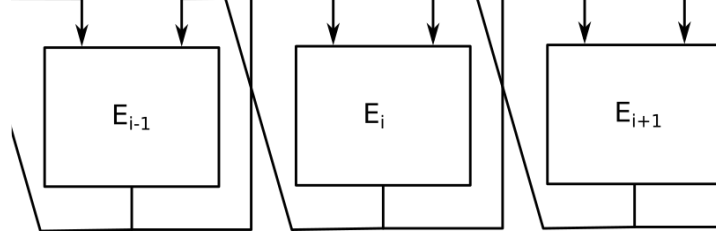


Figure 2.6) Trois cellules d'un automate cellulaire

A chaque coup d'horloge, l'état est mis à jour selon une règle  $R$

$$R : \begin{cases} \{0, 1\}, \{0, 1\}, \{0, 1\} & \longrightarrow \{0, 1\} \\ E_{i-1}, E_i, E_{i+1} & \longmapsto R(E_{i-1}, E_i, E_{i+1}) \end{cases} \quad (2.12)$$

fonction de l'état actuel et de l'état des cellules voisines. Cette règle correspond à une table de correspondance (lookup table, LUT) à trois entrées ce qui explique la très bonne implémentabilité des CA.

Un SPCA est un CA particulier pour lequel les cellules sélectionnent une règle de mise à jour parmi deux disponibles selon la valeur d'un méta-état  $M_i = E_{i+a} \oplus E_{i+b}$  avec  $(a, b) \in \mathbb{N}^2$  et  $\oplus$  l'addition de l'algèbre de Boole. Ainsi la somme des états de deux autres cellules éloignées permet de choisir la règle, ce qui revient à

$$E_i \leftarrow \begin{cases} R_0(E_{i-1}, E_i, E_{i+1}) & \text{si } M_i = 0 \\ R_1(E_{i-1}, E_i, E_{i+1}) & \text{si } M_i = 1 \end{cases} \quad (2.13)$$

avec  $R_0$  et  $R_1$  les deux règles possibles. Quand les deux règles sont bien choisies, l'état et le méta-état de chaque cellule produisent deux bits pseudo-aléatoires par coup d'horloge. Ainsi, un SPCA peut produire à faible coût arithmétique les séquences pseudo-aléatoires nécessaires à la détection.

### 2.3.3 Synthèse du détecteur par parcours d'arborescence stochastique

La figure 2.7 synthétise la détection selon le parcours d'arborescence stochastique et le tableau 2.1 détaille la complexité arithmétique du détecteur en lien avec le pseudo-code. La complexité est évaluée sous les hypothèses suivantes qui visent à améliorer l'implémentabilité sur FPGA.

Tout d'abord, nous contrainsons le choix des paramètres  $k$  et  $\gamma$  pour simplifier certains calculs. Ainsi,  $k$  est supposé être un entier pour que le calcul de la puissance de l'équation (2.11) revienne à une série de multiplications. De plus,  $\gamma$  est fixé à une

### 2.3 Un nouveau détecteur à parcours d'arborescence stochastique

---

```

1 Calculer la décomposition QR de la matrice du canal :  $\mathbf{H} = \mathbf{Q}\mathbf{R}$ 
2 tant que  $\mathbf{H}$  ne change pas faire
3   Appliquer la rotation au vecteur reçu :  $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$ 
4   pour chaque lucioles/chemin faire
5     Initialiser la PED à 0 :  $p \leftarrow 0$ 
6     pour  $k$  de  $N_t$  à 1 faire
7       Calculer l'incrément de distance de chaque enfant avec (1.22)
8       Calculer l'attractivité de chaque enfant avec (2.11)
9       Calculer la probabilité de sélection de chaque enfant avec (2.8)
10      Sélectionner un enfant au hasard selon les probabilités précédentes
11      Mettre à jour la PED en ajoutant un terme à (1.20)
12  si Sortie dure alors
13    retourner le meilleur chemin construit
14  sinon
15    Calculer les LLR avec (1.25) et la liste des chemins construits
16    retourner les LLR

```

---

Figure 2.7) Pseudo-code de la détection par parcours d'arborescence stochastique

---

Étape	Nb $\times$	Nb $+$
Ligne 1) Calculer $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$ .	$N^2$	$(N - 1)N$
À chaque itération de la boucle de la ligne 6	Boucler $N$ fois	
$\hookrightarrow$ Ligne 7) Calculer la métrique de chaque enfant.	$F(N - k - 1 + 2q)$	$F(N - k - 1 + 2q)$
$\hookrightarrow$ Ligne 8) Calculer l'attractivité de chaque enfant.	$Fkq$	0
$\hookrightarrow$ Ligne 9) Calculer la probabilité de chaque enfant.	$Fq$	$Fq$
$\hookrightarrow$ Ligne 10) Sélectionner un enfant au hasard.	0	n/a
$\hookrightarrow$ Ligne 11) Mettre à jour la PED.	$2F$	0
Ligne 20) Approximation max-log si besoin.	$N \log_2(q)$	$N \log_2(q)$
<b>Complexité asymptotique</b>	$\Theta(N^2)$	$\Theta(N^2)$

Tableau 2.1) Complexité arithmétique du détecteur par parcours d'arborescence stochastique selon les instructions de la figure 2.7. On suppose un système MIMO  $N \times N$ , une constellation à  $q$  symboles,  $F$  lucioles/chemins. De plus  $k$  est un entier et  $\gamma$  est une puissance de 2.

puissance de deux pour permettre de calculer la multiplication par  $\gamma$  par des décalages à droite ou à gauche. De plus, la sélection aléatoire des enfants est supposée être obtenue sans produit en lien avec la proposition de la section 2.3.2.

## 2.4 Les références à parcours d'arborescence

Des références issues de la littérature sont nécessaires pour évaluer notre nouveau détecteur à parcours d'arborescence stochastique. Deux algorithmes sont retenus avec l'objectif de représenter la variété des détecteurs par parcours d'arborescence. En cohérence avec l'objectif final du manuscrit, nous ne nous comparerons qu'à des détecteurs qui ont démontré des atouts pour l'implémentation sur FPGA ou ASIC.

Tout d'abord, le K-best dans sa version canonique est sélectionné en tant que fondateur des détecteurs à parcours en largeur (breadth-first search, BFS) qui ont été présentés rapidement en section 1.3.4 [3]. Bien qu'il soit moins optimisé que d'autres détecteurs plus modernes, il reste une référence intéressante car il tolère beaucoup de point de fonctionnement et qu'il ne se règle qu'avec qu'un seul paramètre. Ce détecteur est présenté dans la section 2.4.1. Nous retenons aussi le détecteur best-first parallèle comme représentant des MFS modernes et plus généralement comme l'une des variantes les plus optimisées des parcours d'arborescence [4]. Nous présentons ses caractéristiques en section 2.4.2.

### 2.4.1 Le parcours en largeur canonique : le K-best de [3]

La stratégie de parcours en largeur est avantageuse par sa simplicité et son temps d'exécution indépendant des entrées. De plus, ces détecteurs sont particulièrement adaptés à l'implémentation matérielle et ne sont réglés que par le nombre de chemins conservés  $K$ . A chaque niveau de l'arborescence, les chemins partiels sont prolongés et triés selon leur PED. Seuls les  $K$  meilleurs sont conservés et tous les autres chemins sont élagués. Le procédé est répété au niveau suivant jusqu'à atteindre les feuilles.

La figure 2.8 détaille chaque étape du détecteur en introduisant  $\mathcal{X}_n$  l'ensemble des chemins partiels au niveau  $n \in \llbracket 1, N_t \rrbracket$  et reprenant les  $d_k$  de (1.22). Ainsi, on note  $d_{k,\chi}$  le  $d_k$  correspondant au chemin en construction  $\chi \in \mathcal{X}_n$  et  $\mathbf{d}_\chi$  représente le vecteur contenant tous les  $d_k$ . Enfin, on note  $p_\chi$  la PED du chemin en construction  $\chi$ . Les instructions mettent en évidence les nombreuses parallélisations possibles avec des calculs vectoriels et des itérations de boucles indépendantes.

Le tableau 2.2 évalue la complexité arithmétique de ce détecteur en lien avec le pseudo-code de la figure 2.8. On peut noter que la complexité arithmétique est d'ordre  $\Theta((N-1)N)$  et qu'elle varie linéairement avec le paramètre  $K$ . La référence [3] précise qu'un pré-traitement permet de diviser  $K$  et de réduire la complexité d'un facteur 1 à 5 selon le SNR de fonctionnement. Nous appellerons *mode 1* le fonctionnement sans pré-traitement et *mode 3* celui avec, par cohérence avec cette publication. Les modes 2 et 4 correspondent à des pré-traitements partiels qui ne sont pas détaillés ici. Les simulations seront réalisées avec le *mode 1* par simplicité et un facteur correctif issu de [3] est appliqué *a posteriori* pour obtenir une comparaison équitable. Enfin, le

tableau 2.2 met en lumière que le coût prépondérant est l'évaluation des fonctions objectifs et que le nombre de nœuds visités est donc une bonne métrique pour approximer la complexité.

Enfin la figure 2.9 présente une analyse de Pareto de ce détecteur pour des paramètres  $K$  variant logarithmiquement de 2 à 256 et pour une 16-QAM. Cette analyse prouve la versatilité annoncée en introduction. En effet, bien qu'il ne soit pas le plus optimisé, ce détecteur convient tout aussi bien à des situations visant à réduire la complexité au maximum qu'à une recherche de performance. De plus, cette versatilité est réglée aisément avec un unique paramètre.

### 2.4.2 Le parcours parallèle par métrique de [4]

Le détecteur basé sur un parcours par métrique parallèle de [4] est un détecteur très récent optimisé pour la détection à sorties souples. Il intègre les avantages de nombreux algorithmes à parcours d'arborescence précédents dans une structure optimisée pour l'implémentation sur FPGA et ASIC. Ce détecteur maintient une liste de nœuds pour chaque niveau de l'arborescence. À chaque itération, le meilleur nœud est extrait de chaque liste non vide, prolongé et son meilleur frère et fils sont intégrés dans les listes adéquates. Ce procédé permet d'obtenir un premier chemin complet après  $N_t$  itérations et est extrêmement adapté à l'implémentation matérielle sur FPGA et ASIC. En effet, tous les niveaux de l'arborescence sont indépendants au sein d'une itération. La détection est terminée quand toutes les listes sont vides. La figure 2.10 détaille l'exécution de ce type d'algorithme.

Ce détecteur intègre également plusieurs améliorations inspirées d'autres algorithmes par parcours d'arborescence. Ainsi, chaque liste a une taille pré-définie et les nœuds de plus mauvaises fonctions objectifs sont éliminés si la liste est pleine. Cela permet de limiter la consommation en mémoire à la façon des algorithmes par BFS. De plus, les chemins terminés permettent d'ajuster dynamiquement un rayon pour élaguer au plus tôt les chemins sur la base de leur PED comme le font les algorithmes par séparation et évaluation (branch and bound, BnB).

Contrairement au K-best, le détecteur de [4] ne visite pas toujours le même nombre de nœuds. En effet, les chemins élagués suite à une surcharge des listes et suite à une PED violant la contrainte de rayon, varient d'une entrée à l'autre. Il n'est donc pas possible d'obtenir une formule générale de la complexité de ce détecteur. Le tableau 2.3 donne la complexité arithmétique de chacune des étapes impactantes et la complexité effective sera évaluée par une méthode de Monte Carlo comme décrite en section 2.1.2. Le calcul du rayon (ligne 8 de la figure 2.10) et la mise à jour des contre-hypothèses (ligne 17) ne requièrent que des comparaisons et sont donc négligés. On retrouve une nouvelle fois que le nombre de nœuds visités est une bonne heuristique pour estimer la complexité arithmétique totale.

La figure 2.2 montre que ce détecteur est particulièrement optimisé et qu'il dépasse sans problème le K-best. Cependant, il ne dispose pas de la flexibilité de son concurrent puisque des paramètres extrêmes ne font varier le point de fonctionnement que de quelques dixièmes de dB pour une variation de complexité de moins de 25% [4].

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

---

```

1 Calculer la décomposition :  $\mathbf{QR} = \mathbf{H}$ 
2 tant que  $\mathbf{H}$  ne change pas faire
3   Appliquer la rotation au vecteur reçu :  $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$ 
4   Initialiser toutes les PED à 0 :  $\forall \chi \in \mathcal{X}, p_\chi \leftarrow 0$ 
5   Initialiser tous les  $\mathbf{d}$  pour commencer à évaluer (1.22) :  $\forall \chi \in \mathcal{X}, \mathbf{d}_\chi \leftarrow \tilde{\mathbf{y}}$ 
6   Initialiser l'ensemble des chemins  $\mathcal{X}_{N_t}$  avec la racine
7   pour  $k$  de  $N_t$  à 1 faire
8     Démarrer un ensemble vide pour les chemins étendus :  $\mathcal{X}_{k-1} \leftarrow \emptyset$ 
9     pour chaque chemin partiel  $\chi \in \mathcal{X}_k$  faire
10       pour chaque symbole  $s \in \mathcal{Q}$  faire
11         Étendre le chemin  $\chi$  avec le symbole  $s$  et l'ajouter à  $\mathcal{X}_{k-1}$ 
12         Terminer  $d_{k,\chi}$  selon (1.22) :  $\forall \chi \in \mathcal{X}, d_{k,\chi} \leftarrow d_{k,\chi} - R_{k,k}s$ 
13         Mettre à jour la PED selon (1.20) :  $\forall \chi \in \mathcal{X}, p_\chi \leftarrow p_\chi + d_{k,\chi}^2$ 
14       Trier tous les chemins étendus de  $\mathcal{X}_{k-1}$  selon leur PED
15       Ne conserver que les  $K$  meilleurs chemins étendus dans  $\mathcal{X}_{k-1}$ 
16       Mettre à jour les  $\mathbf{d}$  conservés selon (1.22)
17   si Sortie dure alors
18     retourner le meilleur point de  $\mathcal{X}_1$ 
19   sinon
20     Calculer les LLR avec (1.25) et la liste  $\mathcal{X}_1$ 
21     retourner les LLR

```

---

Figure 2.8) Synopsis de la détection selon l'algorithme K-best de [3]

---

Étape	Nb $\times$	Nb $+$
Ligne 1) Calculer $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$ .	$N^2$	$(N-1)N$
À chaque itération de la boucle de la ligne 7	Boucler $N$ fois	
↪ Ligne 12) Calculer la métrique de chaque enfant.	$Kq$	$Kq$
↪ Ligne 13) Mettre à jour la PED de chaque enfant.	$Kq$	$Kq$
↪ Ligne 16) Mettre à jour l'histoire des chemins.	$K(k-1)$	$K(k-1)$
<b>Sous-total de la boucle</b>	$K \left( 2qN + \frac{(N-1)N}{2} \right)$	
Ligne 20) Approximation max-log si besoin.	$K + N \log_2(q)$	$N \log_2(q)$
<b>Complexité asymptotique</b>	$\Theta(N^2)$	$\Theta(N^2)$

Tableau 2.2) Complexité arithmétique du détecteur K-best selon les instructions de la figure 2.8. On suppose un système MIMO  $N \times N$ , une constellation à  $q$  symboles et on note  $K$  le paramètre de l'algorithme.



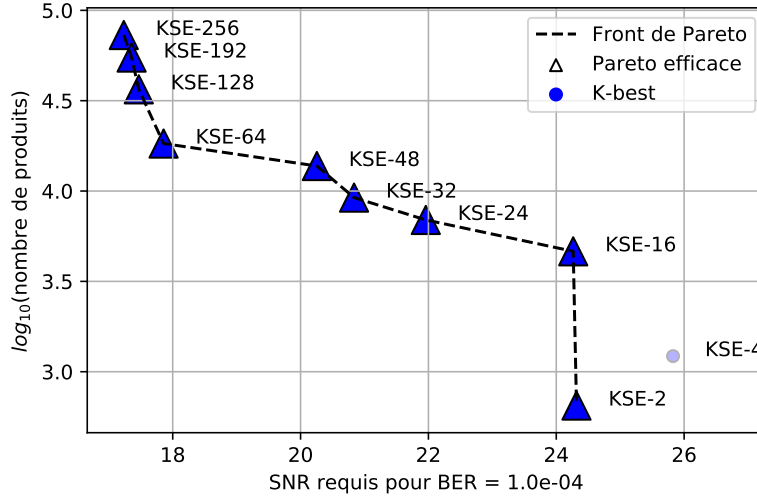


Figure 2.9) Analyse de Pareto du détecteur K-best pour  $K$  variant de 2 à 256 et pour une 16-QAM.

## 2.5 Comparaison du détecteur basé sur le parcours d'arborescence stochastique avec des références

Cette section compare le nouveau détecteur avec les deux références précédemment introduites. Ces trois algorithmes s'appuient tous sur des parcours d'arborescence mais à l'inverse de l'état de l'art, notre détecteur ajoute une composante d'exploration. La section 2.5.1 évalue le surcoût de complexité engendré par l'ajout de l'exploration par comparaison aux méthodes traditionnelles. Par la suite, la section 2.5.2 présente une analyse de Pareto du nouveau détecteur pour différentes modulations afin de le situer par rapport à la littérature.

### 2.5.1 Surcoût de complexité engendré par l'exploration

Avant d'étudier l'éventuelle amélioration des performances que permet l'exploration, il est utile de s'intéresser au surcoût de complexité engendré par cette modification.

Le premier changement majeur pouvant augmenter la complexité est l'ajout d'une composante aléatoire. Cette modification peut engendrer deux problèmes : la nécessité de générer des nombres aléatoires et le fait que la détection ne soit plus déterministe. La section 2.3.2 a présenté deux solutions pour disposer de nombres aléatoires sans surcoûts majeurs à savoir le stockage d'une séquence dans des blocs ROM et le recours à des SPCA pour générer des séquences à la volée. De plus, l'exécution non déterministe pose des complications d'implémentation mais n'est pas rédhibitoire. Le détecteur best-first parallèle est un bon exemple d'algorithme non déterministe mais disposant

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

---

```

1 Calculer la décomposition :  $\mathbf{QR} = \mathbf{H}$ 
2 tant que  $\mathbf{H}$  ne change pas faire
3   Appliquer la rotation au vecteur reçu :  $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$ 
4   Initialiser la liste  $\mathcal{L}_0$  avec les fils de la racine
5   tant que au moins une liste n'est pas vide faire
6     pour chaque liste non vide  $\mathcal{L}_n$  faire
7       Extraire le meilleur nœud  $\chi$  de  $\mathcal{L}_n$ 
8       Calculer le rayon  $r$  correspondant à la plus grande contre-hypothèse susceptible
          d'être impactée
9       si la PED du meilleur frère de  $\chi$  est plus petite que  $r$  alors
10        | Ajouter ce meilleur frère dans la liste  $\mathcal{L}_n$ 
11       Prolonger  $\chi$  pour déterminer son meilleur fils si la PED du meilleur fils de  $\chi$  est
          plus petite que  $r$  alors
12        | Ajouter ce meilleur fils dans la liste  $\mathcal{L}_{n+1}$ 
13       si une nouvelle feuille est présente dans  $\mathcal{L}_{N_t}$  alors
14        | si la nouvelle feuille est la meilleure jamais rencontrée alors
15          | Stocker cette feuille et sa métrique
16        | sinon
17          | Mettre à jour les contre-hypothèses
18       Ne conserver que les meilleurs nœuds de chaque liste
19 retourner les LLR

```

---

Figure 2.10) Synopsis de la détection selon le parcours par métrique parallèle de [4]

---

Étape	Nb ×	Nb +
Ligne 1) Calculer $\tilde{\mathbf{y}} = \mathbf{Q}^T \mathbf{y}$	$N^2$	$(N - 1)N$
Ligne 4 & 11) Prolonger un nœud au niveau $n$	$(N - n + 1)q$	$2q$
Ligne 19) Calculer les LLR	$Nq$	$Nq$
<b>Complexité asymptotique</b>	$\Theta(N^2)$	$\Theta(N^2)$

Tableau 2.3) Complexité arithmétique du détecteur best-first parallèle selon les instructions de la figure 2.10. On suppose un système MIMO  $N \times N$  et une constellation à  $q$  symboles.

## 2.5 Comparaison du parcours d'arborescence stochastique avec des références

d'implémentations très performantes. Ainsi, l'exploration aléatoire ne devrait pas engendrer de surcoût rédhibitoire.

Le deuxième changement significatif de la nouvelle méthode est la construction indépendante des  $F$  chemins. En effet, les détecteurs de la littérature construisent tous les chemins au sein d'une stratégie générale qui les lie les uns aux autres. Par exemple, la détection par BFS prolonge tous les chemins en même temps pour déterminer les branches les plus intéressantes. De même, le détecteur best-first hiérarchise les chemins partiels les uns par rapport aux autres afin de sélectionner les prolongations prometteuses. En opposition avec ces stratégies, les  $F$  chemins de notre nouveau détecteur sont construits indépendamment les uns des autres. Ainsi, un même nœud peut être visité plusieurs fois par différentes lucioles/chemins. Or, nous avons établi que la complexité totale est liée au nombre de nœuds visités. Visiter plusieurs fois le même nœud peut donc avoir un impact significatif.

La figure 2.11 analyse le nombre de nœuds visités plusieurs fois au cours d'une même détection. Pour cela, nous représentons le ratio du nombre de nœuds uniques visités sur le nombre total de nœuds visités. Le cas étudié est une détection souple sur un canal 4x4 avec un SNR de 20 dB et une 16-QAM. Nous fixons les paramètres à des valeurs typiques :  $k = 2$ ,  $F$  et  $\gamma$  varient entre deux valeurs. Les deux boîtes de Tukey supérieures analysent l'impact d'un changement du nombre de lucioles/chemins à  $\gamma$  fixé. Les deux boîtes inférieures analysent réciproquement l'impact du changement de  $\gamma$  à  $F$  constant.

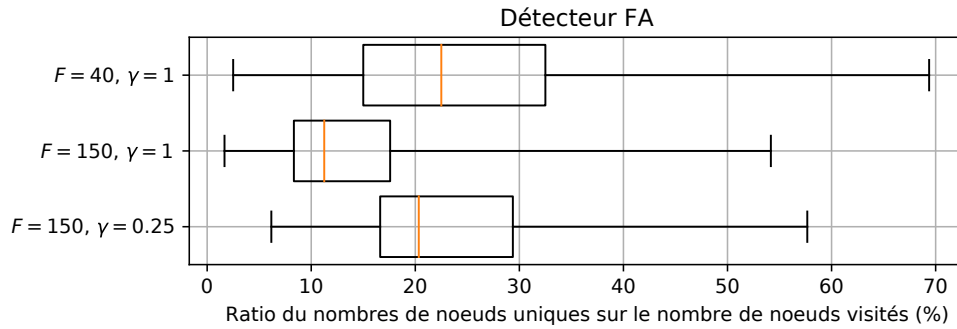


Figure 2.11) Statistiques des nœuds visités pour un système 4x4, une 16-QAM, un SNR de 20 dB et  $k = 2$ . Les moustaches représentent les valeurs minimales et maximales rencontrées.

La partie supérieure du graphe indique que multiplier par quatre le nombre de lucioles/chemins revient à diviser par deux le ratio. Ainsi, construire plus de chemins engendre forcément la visite de nouveaux nœuds uniques mais l'augmentation du nombre total de nœuds est suffisante pour faire diminuer le ratio. Cela signifie donc que beaucoup de nouveaux chemins sont au moins partiellement identiques à ceux déjà construits. La partie inférieure présente la variation du ratio en divisant par 4 la valeur de  $\gamma$  pour un nombre de lucioles/chemins constant. Dans ce cas de figure, le ratio augmente significativement ce qui est cohérent avec l'analyse présentée en section 2.3.1

sur le sens de  $\gamma$ . En effet, ce paramètre règle le compromis exploration-exploitation : plus il est faible, plus le détecteur explore. Ainsi, il est logique que le nombre de nœuds uniques soit beaucoup plus important quand l'algorithme explore d'avantage.

Enfin, les trois statistiques s'accordent pour montrer que le ratio reste faible dans toutes les circonstances. Ainsi, la médiane ne dépasse jamais les 20 % et le troisième quartile reste sous les 35 % même dans le pire des cas. Or le nombre de nœuds visités est une bonne estimation de la complexité de l'algorithme et visiter plusieurs fois le même nœud est donc très coûteux. Par conséquent, la suite du manuscrit suppose qu'à chaque visite d'un nouveau nœud, sa PED est stockée dans un bloc RAM pour qu'une nouvelle visite du nœud ne coûte qu'une recherche dans une table. Cette mémoire évite donc de répéter des calculs coûteux similaires. Les statistiques montrent que l'entretien d'une telle liste permet d'économiser plus de 75 % des calculs de fonction objectif ce qui est largement suffisant pour justifier cette nouvelle contrainte matérielle.

### 2.5.2 Comparaison des performances au sens de Pareto

Des tests préliminaires montrent que notre détecteur ne fonctionne pas pour les QPSK. Cependant, trouver au moins un cas d'application réaliste à cette nouvelle détection est suffisant pour démontrer la pertinence de l'exploration. Ainsi, cette section se concentre exclusivement sur les 16-QAM pour lesquelles les résultats préliminaires sont favorables.

#### Définition de l'ensemble des paramètres d'études

L'analyse de Pareto doit parcourir l'ensemble des réglages possibles de chaque détecteur pour être objective. Le tableau 2.4 liste l'ensemble des paramètres testés pour chacun des détecteurs. Le nombre de chemins survivants du K-best est fixé par un balayage logarithmique depuis le  $K$  minimal jusqu'à celui au-delà duquel les performances ne s'améliorent plus. Ainsi, on peut voir sur le Pareto de la figure 2.9 une saturation des performances vis-à-vis du paramètre  $K$ . Les tailles des listes du best-first sont celles proposées dans [4]. Simuler plus de taille n'est pas pertinent puisque ces tailles ne changent que très peu le point de fonctionnement.

Nous parcourons l'ensemble des trois paramètres de notre nouveau détecteur : le nombre de lucioles/chemins  $F$  et les paramètres d'exploration  $k$  et  $\gamma$ . Le nombre de lucioles/chemins est balayé de 10 jusqu'à une position où la complexité est si élevée que le détecteur ne sera jamais efficace au sens de Pareto. Les deux paramètres d'exploration sont balayés sous les contraintes permettant de simplifier les calculs :  $k$  est un entier et  $\gamma$  une puissance de 2. Encore une fois, les plages de valeurs sont étendues jusqu'à atteindre des zones de fonctionnement ne pouvant pas être efficaces.

#### Analyse de Pareto sur l'ensemble des paramètres

La figure 2.12 présente l'analyse de Pareto des trois détecteurs pour une 16-QAM. Chaque configuration est représentée par un point de fonctionnement. Les configurations efficaces au sens de Pareto sont représentées par des triangles, les autres par

## 2.5 Comparaison du parcours d'arborescence stochastique avec des références

Détecteur	Paramètre	Valeurs testées
K-Best	$K$	2, 4, 8, 12, 16, 24, 32, 48, 64, 128, 192, 256
Best-first	Taille des listes	(1, 3, 5), (1, 7, 10), (1, 12, 15)
Nouveau FA	$k$	1, 2, 3, 4, 5
	$\gamma$	$2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1$
	$F$	10, 20, 30, 40, 60, 90, 130, 150, 175, 200, 250, 275

Tableau 2.4) Liste des paramètres testés dans l'analyse de Pareto

des disques. La couleur représente le type de détecteur. Notre nouveau détecteur est beaucoup plus présent puisqu'il possède trois paramètres et a donc significativement plus de configurations possibles.

L'analyse de Pareto montre que le détecteur à parcours d'arborescence stochastique domine les deux références sur la quasi-totalité des points de fonctionnement. Ainsi le nouveau détecteur atteint les mêmes performances que le K-best *mode 3* de [3] avec 10 fois moins de produits par vecteurs détectés. Ce résultat est valable pour les points de fonctionnement compris entre 17 dB et 23 dB. De plus, le nouveau détecteur à exploration domine toutes les configurations du récent parcours par métrique de [4]. Pour autant, le K-best reste efficace au sens de Pareto lorsque l'on recherche une détection la plus simple possible quitte à ce que cela nécessite un SNR très élevé. En effet, l'approche aléatoire de notre détecteur requiert un nombre minimum de lucioles/chemins pour être efficace ce qui explique qu'il ne peut pas atteindre de point de fonctionnement à complexité aussi faible que le K-best avec 2 chemins.

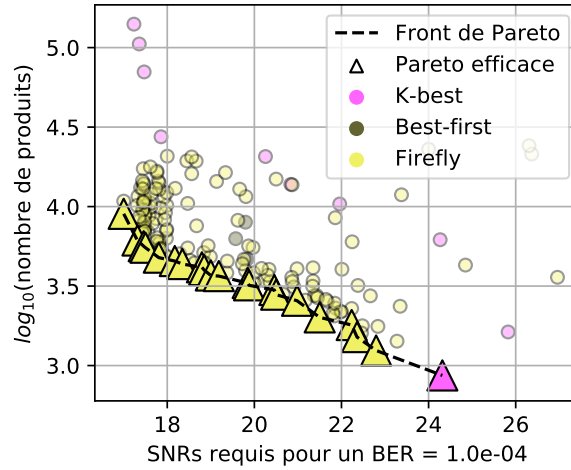


Figure 2.12) Analyse de Pareto des trois détecteurs pour une 16-QAM avec  $N = 4$ .

### Réduction du nombre de paramètres de réglages

La grande flexibilité du détecteur à parcours d'arborescence stochastique est en partie expliquée par ses trois paramètres qui laissent une plus grande marge de réglage. Cependant, un nombre de paramètres trop élevé peut aussi être un défaut pour deux raisons. Tout d'abord, plus un détecteur possède de paramètres, plus il est difficile à régler. De plus, le réglage des paramètres est un frein à l'implémentation de détecteurs versatiles. En effet, il faut alors prévoir une architecture capable de se configurer pour fonctionner selon chaque réglage ce qui implique des surcoûts de ressources. Ces raisons nous poussent à réduire le nombre de paramètres de réglage de notre détecteur.

La figure 2.13 présente deux nouvelles analyses de Pareto avec les mêmes références dans les mêmes conditions. Cependant, la valeur de  $k$  est fixée respectivement à 2 et 3. Ces deux graphes montrent que notre détecteur ne perd ni flexibilité, ni efficacité en fixant ce paramètre à une valeur adéquate. Le nombre de paramètres de réglage peut être réduit à seulement deux :  $\gamma$  et  $F$ .

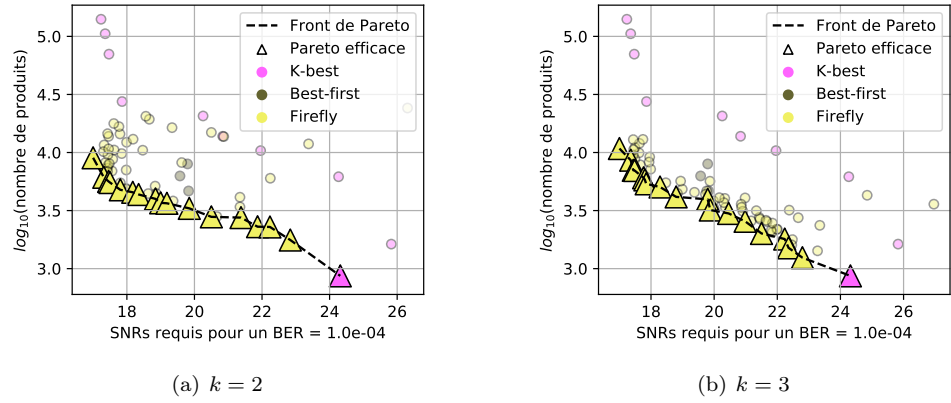


Figure 2.13) Analyse de Pareto en fixant le paramètre  $k$  de notre détecteur.

Il est possible de réduire encore l'espace de réglage de notre détecteur en imposant une contrainte sur les valeurs possibles de  $\gamma$ . En effet, la plupart des réglages efficaces au sens de Pareto dans le cas  $k = 2$  partagent des valeurs de  $\gamma$  comprises entre 0.25 et 1. Ainsi, nous proposons de contraindre  $\gamma$  à prendre ses valeurs parmi trois valeurs différentes. La figure 2.14 présente une nouvelle analyse de Pareto pour  $k = 2$  sous la contrainte  $\gamma \in \{0.25, 0.5, 1\}$ . Seul un point du front de Pareto de la figure 2.13(a) manque sur la nouvelle version. Il s'agit du point de fonctionnement le plus extrême visant le SNR minimal à forte complexité. Pour autant, le nouveau front montre que notre détecteur reste versatile et efficace au sens de Pareto. De plus, le nouveau point extrême ne perd que 0.3 dB par rapport au réglage inaccessible qui requiert  $\gamma = 0.125$ .

Le détecteur à parcours d'arborescence stochastique peut donc fournir des performances particulièrement intéressantes avec deux paramètres seulement :  $\gamma$  à choisir parmi trois valeurs possibles et le nombre de lucioles/chemins  $F$  laissé libre. De plus

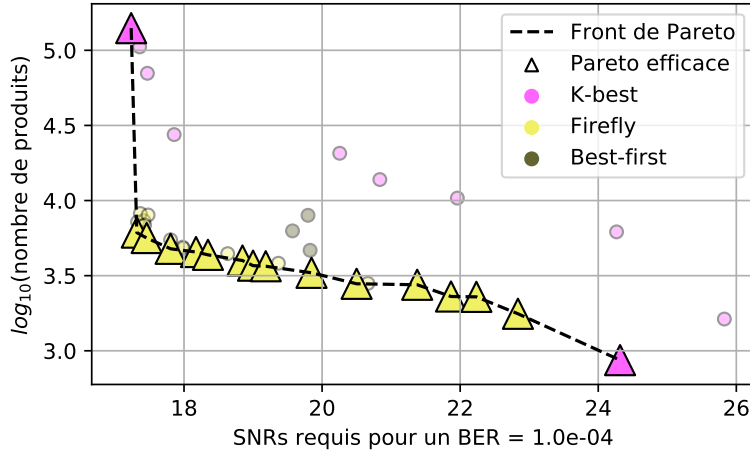


Figure 2.14) Analyse de Pareto en fixant  $k = 2$  et en imposant  $\gamma \in \{0.25, 0.5, 1\}$

un réglage supplémentaire permettant de gagner 0.3 dB est accessible en autorisant une quatrième valeur pour  $\gamma$ .

## 2.6 Conclusion sur la détection par parcours d'arborescence stochastique

Ce chapitre a testé la validité de l'hypothèse relevée suite à l'état de l'art : *les détecteurs exploitant le modèle deviennent-ils plus résilients après ajout d'un comportement exploratoire ?* Pour l'évaluer, un nouveau détecteur a été développé à l'intersection des stratégies par parcours d'arborescence et des métaheuristiques bio-inspirées. Ainsi, notre nouveau algorithme exploite le modèle de la même façon que les classiques K-best et best-first. Cependant, il se distingue de ces références par son exploration aléatoire inspirée du FA. La comparaison de notre détecteur avec les autres références permet donc d'évaluer l'intérêt de leur seule différence : la présence ou non d'exploration.

Trois méthodes de comparaison sont retenues pour comparer notre nouveau détecteur aux références. Tout d'abord, nous reprenons les deux métriques usuelles les plus évidentes : l'évolution du BER en fonction du SNR et la complexité de calcul. La complexité retenue est la complexité arithmétique en nombre de produits car il s'agit d'une métrique approchant assez bien le débit et la consommation énergétique d'une implémentation matérielle efficace. De plus, ces deux critères sont mis en lien par des analyses de Pareto afin d'étudier objectivement le compromis entre les deux métriques antagonistes.

Notre nouvel algorithme est développé en rapprochant le FA du paradigme par parcours d'arborescence. Le FA est originellement une métaheuristique d'optimisation par évolution de population caractérisée par le déplacement de lucioles selon la fonction

## 2 Étude d'un détecteur à parcours d'arborescence stochastique

objectif superposée à une exploration aléatoire. Nous avons montré dans ce chapitre ainsi que dans [5] que le détecteur exploitant le FA de [2] peut être réinterprété comme un parcours d'arborescence stochastique ce qui en fait une inspiration parfaite pour l'objectif que nous nous sommes fixé.

Nous avons amélioré le détecteur de [2] pour pouvoir le comparer avec les parcours d'arborescence sans exploitation de la littérature. En effet, la version entièrement bio-inspirée de [2] est restreinte à la détection dure des symboles d'une QPSK. La section 2.3.2 a permis d'étendre le fonctionnement à toutes les modulations, et en particulier à toutes les QAM. Une détection à sorties souples a également été proposée en évaluant l'approximation max-log sur une liste de chemins. Cette méthode de production des LLR est identique aux références ce qui évite de biaiser la comparaison finale. Enfin, la complexité a été évaluée et réduite à son maximum. Pour cela, nous avons simplifié le calcul de l'exponentielle par un approximant de Padé et recensé plusieurs méthodes permettant de produire des nombres aléatoires sans produit. Toutes ces approximations permettent d'adapter notre nouveau détecteur à la cible matérielle FPGA et ASIC.

La version optimisée a été comparée aux détecteurs de références dans la section 2.5 selon les trois critères précédents. Nous avons d'abord évalué le surcoût de complexité induit par l'exploration aléatoire en décomptant le nombre de nœuds visités au cours d'une détection typique. Nous avons pu montrer qu'un système de cache évitant de recalculer plusieurs fois les mêmes nœuds permet de réduire de plus 75 % les calculs de fonctions objectifs. Ainsi, l'exploration aléatoire ne semble pas engendrer de surcoût rédhibitoire. De plus, les comparaisons de Pareto montrent que notre détecteur est efficace sur la quasi-totalité des points de fonctionnement pour une 16-QAM. Il démontre une versatilité largement supérieure au best-first tout en divisant d'un facteur proche de dix la complexité arithmétique comparé au K-best. Enfin, nous avons réduit le nombre de paramètres de réglage en préservant les performances et la versatilité pour simplifier l'implémentation.

Ces comparaisons vont dans le sens de notre hypothèse puisque l'ajout d'exploration aléatoire a bien permis d'améliorer sensiblement les performances. Cette conclusion motive le développement d'un détecteur intégrant l'exploration au sein même de sa conception qui sera abordé dans les deux derniers chapitres. En effet, notre détecteur à parcours d'arborescence stochastique n'utilise l'exploration que comme un ajout sur une stratégie pré-existante. Nous cherchons donc dans la suite du manuscrit à développer un détecteur intégrant dans sa stratégie même le compromis exploration-exploitation.



## 3 Diagnostic du détecteur géométrique historique

Le chapitre précédent a évalué l'hypothèse qui suppose que l'exploration rend les détecteurs plus résilients. Pour ce faire, un détecteur à parcours d'arborescence pourvu d'une composante d'exploration a été comparé à des références sans exploration basées sur la même stratégie. L'analyse de Pareto a montré que notre détecteur était plus efficace que les détecteurs de références, ce qui va dans le sens de l'hypothèse.

Dans l'étude précédente, l'exploration était un ajout à une stratégie pré-existante. Cette approche était nécessaire car l'évaluation de l'hypothèse requerrait que la présence d'exploration soit la seule différence avec les références. Ce chapitre s'intéresse cette fois à un détecteur intégrant dès sa conception le compromis exploration-exploitation, à savoir le détecteur géométrique historique de [6, 7]. Contrairement au détecteur à parcours d'arbre stochastique du chapitre précédent, l'exploration fait cette fois partie intégrante de la stratégie. Ainsi, les bénéfices de l'exploration devraient être plus présents que lorsqu'elle est un ajout tardif.

La section 3.1 introduit toutes les notions mathématiques nécessaires à la compréhension et l'étude de cette classe de détecteur. En particulier, une présentation détaillée de la décomposition en valeurs singulières (singular values decomposition, SVD) et de ses propriétés utiles est fournie en section 3.1.2. Par la suite, la section 3.2 expose la stratégie géométrique avant qu'elle ne soit évaluée dans la section 3.3. Plus précisément, les sections 3.3.1 et 3.3.2 quantifient la complexité et les performances par rapport à un détecteur de la littérature et la section 3.3.3 présente un prototype sur système sur puce (system on chip, SoC) pour diagnostiquer l'implémentabilité matérielle de la stratégie. Enfin, la section 3.4 synthétise les atouts et les défauts du détecteur géométrique historique pour amorcer le travail d'amélioration du chapitre suivant.

### 3.1 Préambule mathématique

Cette section introduit les notions mathématiques nécessaires à la présentation et l'étude des détecteurs géométriques. Ainsi, la section 3.1.1 exprime le modèle de transmission en variables réelles et la section 3.1.2 détaille la réécriture de la fonction objectif qui a été déjà présentée rapidement dans la section 1.3.7.

### 3.1.1 Expression du modèle de transmission dans le domaine réel

Le détecteur géométrique exploite le modèle (1.9) sous sa forme réelle. Ainsi, nous définissons de nouvelles variables réelles qui conservent leur sens précédent :

$$\mathbf{y}^{\Re} \triangleq \begin{pmatrix} \Re(\mathbf{y}) \\ \Im(\mathbf{y}) \end{pmatrix}, \quad (3.1)$$

$$\mathbf{H}^{\Re} \triangleq \begin{pmatrix} \Re(\mathbf{H}) & -\Im(\mathbf{H}) \\ \Im(\mathbf{H}) & \Re(\mathbf{H}) \end{pmatrix}, \quad (3.2)$$

$$\mathbf{x}^{\Re} \triangleq \begin{pmatrix} \Re(\mathbf{x}) \\ \Im(\mathbf{x}) \end{pmatrix}, \quad (3.3)$$

$$\mathbf{w}^{\Re} \triangleq \begin{pmatrix} \Re(\mathbf{w}) \\ \Im(\mathbf{w}) \end{pmatrix}, \quad (3.4)$$

$$\mathcal{Q}^{\Re} \triangleq \Re(\mathcal{Q}), \quad (3.5)$$

$$N_t^{\Re} \triangleq 2N_t, \quad (3.6)$$

$$N_r^{\Re} \triangleq 2N_r. \quad (3.7)$$

Avec ces notations, le modèle (1.9) trouve son équivalent direct en écriture réelle :

$$\mathbf{y}^{\Re} = \mathbf{H}^{\Re} \mathbf{x}^{\Re} + \mathbf{w}^{\Re}. \quad (3.8)$$

En termes de traitement de signal, le passage d'un modèle complexe à un réel revient à traiter indépendamment les composantes réelles et imaginaires de la modulation. Mathématiquement, il s'agit d'exprimer le même espace vectoriel en utilisant  $\mathbb{R}$  plutôt que  $\mathbb{C}$ . Ces deux interprétations confirment que le nouveau modèle contient exactement la même quantité d'informations que le modèle précédent. Ainsi, la détection n'est ni plus complexe, ni plus simple, avec un modèle ou un autre. Comme la fin du manuscrit n'utilise que le modèle réel et qu'il n'y a pas de risque de confusion, nous omettrons l'exposant  $\Re$  par simplicité de lecture. Dans le cas où il faudrait faire référence à une variable du modèle complexe, celle-ci sera notée avec un exposant  $\Im$ . Par exemple,  $\mathcal{Q}^{\Im}$  représente la constellation à valeur complexe.

### 3.1.2 Prétraitement par décomposition en valeurs singulières

Les détecteurs géométriques exploitent un pré-traitement s'appuyant sur la décomposition en valeurs singulières (singular values decomposition, SVD) qui a déjà été rapidement discuté en section 1.3.7. Cette section définit cette décomposition et explicite son lien avec la fonction objectif.

#### Définition et calcul de la décomposition en valeurs singulières

Sur le corps des réels, la SVD exprime que pour toute matrice  $\mathbf{H} \in \mathbb{R}^{N_r \times N_t}$ , il existe deux matrices orthogonales  $\mathbf{U}$  et  $\mathbf{V}$  telles que  $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$  avec  $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$  une matrice

diagonale et  $n \triangleq \min(N_t, N_r)$ . Les coefficients  $\{\lambda_i : i \in \llbracket 1, n \rrbracket\}$  de  $\mathbf{\Lambda}$  sont appelées valeurs singulières et  $\mathbf{U}$  et  $\mathbf{V}$  peuvent être choisies telles que

$$0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n. \quad (3.9)$$

Avec ces définitions et notations, le nombre de valeurs singulières non nulles est égal au rang de la matrice décomposée et la norme de la matrice peut s'écrire

$$\|\mathbf{H}\|_F^2 = \sum_{i=1}^n \lambda_i^2. \quad (3.10)$$

La SVD peut être calculée de différentes façons et possède des implémentations dans la grande majorité des langages de programmation. Le choix des algorithmes dépend de la forme de la matrice  $\mathbf{H}$  ainsi que des sorties recherchées. Certains algorithmes sont, par exemple, plus adaptés aux matrices carrées [76] quand d'autres sont préférables pour les matrices possédant une dimension bien plus grande que l'autre. De même, des simplifications sont possibles si les matrices  $\mathbf{U}$  et/ou  $\mathbf{V}$  ne sont pas nécessaires [77]. Pour une comparaison de ces algorithmes, voir [78, Section 8.6]. Dans tous les cas, le nombre d'opérations pour obtenir la SVD croît cubiquement avec la plus grande dimension de la matrice du canal  $\mathbf{H}$ .

### Décomposition en valeurs singulières et fonction objectif

La section 1.3.4 développait le pré-traitement par décomposition QR et la réécriture de la fonction objectif associée. Cette section présente le pré-traitement analogue reposant sur la SVD. Ce processus a déjà été présenté en section 1.3.7 mais il est repris ici pour permettre une lecture linéaire du chapitre.

En accord avec (1.11), nous définissons  $\mathbf{x}_0 \triangleq \mathbf{H}^+ \mathbf{y}$  avec  $\mathbf{H}^+$  la pseudo-inverse de  $\mathbf{H}$ . La fonction objectif peut alors être ré-exprimée à partir des différentes matrices :

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{H}(\mathbf{x}_0 - \mathbf{x})\|^2 \quad (3.11)$$

$$= (\mathbf{H}(\mathbf{x}_0 - \mathbf{x}))^T \mathbf{H}(\mathbf{x}_0 - \mathbf{x}) \quad (3.12)$$

$$= (\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T(\mathbf{x}_0 - \mathbf{x}))^T \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T(\mathbf{x}_0 - \mathbf{x}) \quad (3.13)$$

$$= (\mathbf{V}^T(\mathbf{x} - \mathbf{x}_0))^T \mathbf{\Lambda}\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}(\mathbf{V}^T(\mathbf{x} - \mathbf{x}_0)) \quad (3.14)$$

$$= (\mathbf{V}^T(\mathbf{x} - \mathbf{x}_0))^T \mathbf{\Lambda}^2 (\mathbf{V}^T(\mathbf{x} - \mathbf{x}_0)) \quad (3.15)$$

en utilisant l'orthogonalité de  $\mathbf{U}$  et la diagonalité de  $\mathbf{\Lambda}$ .

Soit la famille  $\{\mathbf{v}_i : i \in \llbracket 1, N_t \rrbracket\}$  des colonnes de  $\mathbf{V}$ . Cette famille forme une base de l'espace puisque  $\mathbf{V}$  est orthogonale. Soient  $\{\alpha_i \in \mathbb{R} : \llbracket 1, N_t \rrbracket\}$  les coordonnées du vecteur  $\mathbf{x} - \mathbf{x}_0$  dans cette base. La fonction objectif peut alors s'écrire comme une fonction quadratique dans la nouvelle base :

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \sum_{i=1}^{N_t} \alpha_i^2 \lambda_i^2. \quad (3.16)$$

### Isovaleurs de la fonction objectif

Soit  $\mathcal{I}_\nu$  l'isovaleur définie par

$$\mathcal{I}_\nu = \left\{ \mathbf{x} : \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \sum_{i=1}^{N_t} \alpha_i^2 \lambda_i^2 = \nu \right\} \quad (3.17)$$

avec  $\nu \in \mathbb{R}_+$  une constante.

Nous supposons dans la suite que toutes les valeurs singulières sont non nulles (*ie*,  $\mathbf{H}$  est de rang complet), ce qui est typiquement le cas [79]. La seconde expression de la fonction objectif montre alors que  $\mathcal{I}_\nu$  est une hyper-ellipsoïde dont les demi-axes sont de la forme  $\nu/\lambda_i^2$ . De plus, les vecteurs directeurs des demi-axes sont les colonnes de la matrice  $\mathbf{V}$  qui forment une base orthogonale. Enfin, le centre de l'hyper-ellipsoïde est  $\mathbf{x}_0 = \mathbf{H}^+\mathbf{y}$ . La figure 3.1(a) illustre dans le plan ces propriétés avec une isovaleur en pointillé ainsi que les deux vecteurs  $\mathbf{v}_i/\lambda_i$  en bleu.

Puisque l'équation (1.1) fixe la valeur de  $\|\mathbf{H}\|_F^2$  pour normaliser le canal et que (3.10) lie cette norme aux valeurs singulières, nous savons que ces dernières ne sont pas totalement indépendantes. Pour autant, il est possible qu'une ou plusieurs valeurs singulières deviennent prépondérantes devant les autres ce qui signifie que certains demi-axes  $\nu/\lambda_i^2$  deviennent négligeables. Ainsi, l'hyper-ellipsoïde s'applatit suffisamment pour considérer qu'elle a perdu une ou plusieurs dimensions. On parle alors d'une matrice de canal  $\mathbf{H}$  mal conditionnée. Ce cas de figure est représenté sur la figure 3.1(b).

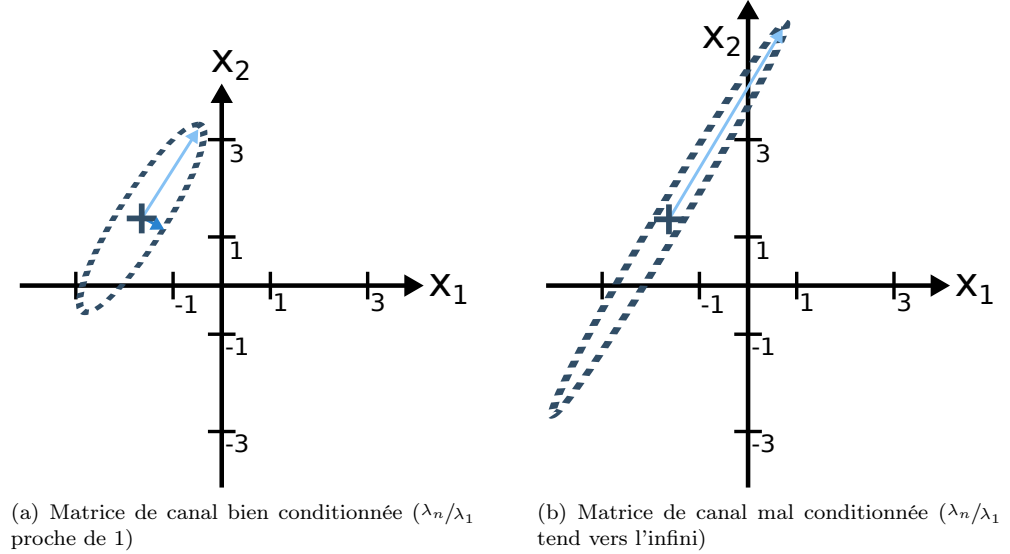


Figure 3.1) Lien entre les valeurs et vecteurs singuliers et les isovaleurs de la fonction objectif.

### 3.2 La détection géométrique historique

Les équations (1.1) et (3.10) impliquent que si l'une des valeurs singulières grandit, les autres doivent diminuer en conséquence. Ainsi, les cas les mieux conditionnés produisent des demi-axes de valeurs équivalentes. À l'inverse, les cas mal conditionnés impliquent que les isovaleurs sont aplaties dans certaines directions et très étendues dans les autres.

#### 3.1.3 Obtention de la pseudo-inverse de Moore-Penrose

La pseudo-inverse est une généralisation de l'inverse aux matrices non-inversibles et/ou carrées. Pour des réels, la pseudo-inverse de  $\mathbf{H} \in \mathbb{R}^{N_r \times N_t}$ , notée  $\mathbf{H}^+ \in \mathbb{R}^{N_t \times N_r}$ , est définie comme l'unique matrice vérifiant les quatre *conditions de Moore-Penrose* [78, Section 5.5.2] :

$$\begin{cases} \mathbf{H}\mathbf{H}^+\mathbf{H} &= \mathbf{H}, \\ \mathbf{H}^+\mathbf{H}\mathbf{H}^+ &= \mathbf{H}^+, \\ (\mathbf{H}\mathbf{H}^+)^T &= \mathbf{H}\mathbf{H}^+, \\ (\mathbf{H}^+\mathbf{H})^T &= \mathbf{H}^+\mathbf{H}. \end{cases} \quad (3.18)$$

La pseudo-inverse recouvre l'inverse usuelle pour des matrices carrées de rang complet.

Pour les matrices de rang complet, la pseudo-inverse peut-être obtenue par

$$\mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \quad (3.19)$$

qui vérifie bien les quatre conditions. L'obtention de la pseudo-inverse par cette méthode requiert deux produits matriciels et une inversion matricielle.

Cependant, la SVD permet d'obtenir plus efficacement la pseudo-inverse puisque  $\mathbf{H}^+ = \mathbf{V}\mathbf{\Lambda}^+\mathbf{U}^T$  vérifie également les conditions avec

$$\mathbf{\Lambda}^+ = \text{diag} \left( \frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \dots, \frac{1}{\lambda_r}, 0, \dots, 0 \right) \quad (3.20)$$

et  $r$  le rang de  $\mathbf{H}$ . Ainsi, la connaissance de la SVD remplace une inversion matricielle par l'inversion de  $r$  scalaires ce qui est bien plus simple. De plus, cette méthode a l'avantage de ne pas avoir besoin de l'hypothèse  $\mathbf{H}$  de rang complet. Ainsi, le surcoût induit par le calcul d'une pseudo-inverse en sus d'une SVD est assez réduit.

## 3.2 La détection géométrique historique

La section 1.3.7 de l'état de l'art a rapidement présenté le détecteur géométrique dans sa version historique [6, 7]. La détection géométrique à sorties souples repose sur un processus en deux étapes qui vise à construire une liste de solutions possibles  $\mathcal{S} \subset \mathcal{Q}^{N_t}$ . Cette liste est ensuite utilisée conjointement à l'approximation max-log pour produire les LLR selon l'équation (1.25). Les deux étapes de l'algorithme opèrent à deux niveaux différents : une recherche grossière à l'échelle de l'espace total puis une recherche fine autour des points prometteurs. Cette section détaille en profondeur le fonctionnement de cette version et met en avant son caractère exploratoire.

### 3.2.1 Recherche générale : exploration géométrique

L'objectif de l'étape de recherche générale est de sélectionner un ensemble  $\mathcal{P} \subset \mathcal{Q}^{N_t}$  de points prometteurs qui seront raffinés dans la seconde étape. La construction de cet ensemble poursuit des critères antagonistes :

1. les points de  $\mathcal{P}$  doivent être prometteurs, c'est à dire avoir une fonction objectif faible ;
2. les points de  $\mathcal{P}$  doivent être aussi différents que possible ;
3. le cardinal  $p \triangleq \text{card}(\mathcal{P})$  doit être le plus faible possible.

Le premier critère assure que la future recherche locale se concentrera sur des zones de l'espace où la solution est susceptible de se trouver. Le second critère garantit que les recherches fines investigueront bien des zones différentes de l'espace. Enfin, le troisième critère réduit le nombre de recherches locales à venir pour éviter une explosion de la complexité arithmétique. L'expression de la fonction objectif (3.16) montre bien l'antagonisme entre les deux premiers critères. En effet, des points très éloignés auront des coordonnées très différentes ce qui induira forcément une forte fonction objectif pour au moins certains d'entre eux.

L'exploration historique s'appuie sur une interprétation géométrique de la fonction objectif pour produire l'ensemble  $\mathcal{P}$ . Pour ce faire, nous définissons les droites  $\Delta_i$  passant par  $\mathbf{x}_0$  et dirigées par les  $\mathbf{v}_i$ . L'expression (3.16) ainsi que l'ordre croissant des  $\lambda_i$  (Cf. (3.9)) met en évidence que la fonction objectif croît moins vite lorsque l'on se déplace le long des premières droites que le long des dernières. Ainsi, la sélection de différents points proches des premières droites valide les critères fixés. La figure 3.2 illustre ce propos dans une coupe en 2 dimensions pour une QPSK. On y retrouve bien que les points le long de la droite violette  $\Delta_1$  valident à la fois les critères de différence et de qualité. En effet, les deux points proches de  $\Delta_1$  forment le couple de points le plus différent ayant une fonction objectif totale faible.

Plusieurs méthodes ont été proposées pour sélectionner des points le long des premières droites [6], toutes reposant sur des intersections entre les  $\Delta_i$  et d'autres éléments géométriques. Nous présentons ici celle qui a été retenue dans notre travail préliminaire en raison de sa bonne implémentabilité [8]. L'intersection des premières  $\Delta_i$  est effectuée avec chaque hyperplan de base  $\mathcal{H}_i = \{\mathbf{x} \in \mathbb{R}^{N_t} : x_i = 0\}$ . Par la suite, le détecteur recherche le point de la constellation le plus proche de chaque intersection. En cas d'égalité, le point le plus proche de  $\mathbf{x}_0$  est retenu. La figure 3.3 illustre ce processus dans le même exemple que la figure 3.2.

La méthode précédemment décrite permet de sélectionner des points qui remplissent les deux premiers critères mais elle produit des ensembles  $\mathcal{P}$  de trop grande taille. C'est pourquoi, le détecteur ne conserve que  $C$  candidats par droite  $\Delta_i$  et ne traite que les  $D$  premières droites. Ainsi,  $p \triangleq \text{card}(\mathcal{P}) = CD$  par construction ce qui permet de réduire le cardinal de  $\mathcal{P}$  et donc de remplir le troisième critère.

### 3.2 La détection géométrique historique

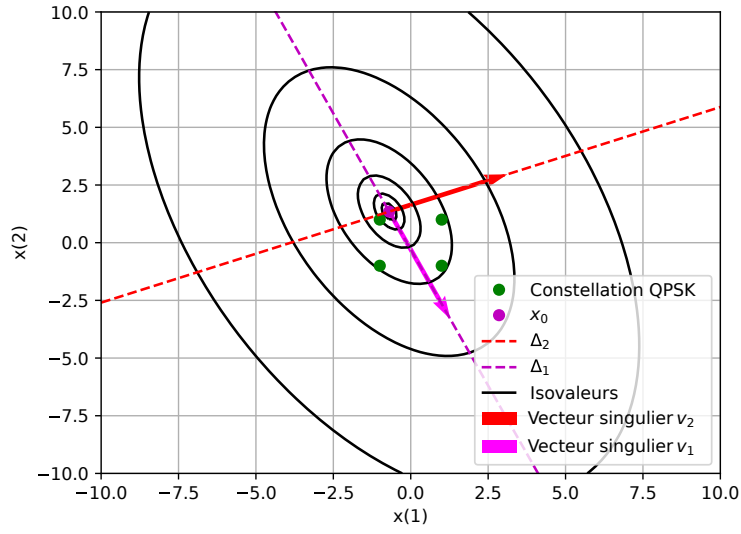


Figure 3.2) Variation de la fonction objectif le long des  $\Delta_i$ .

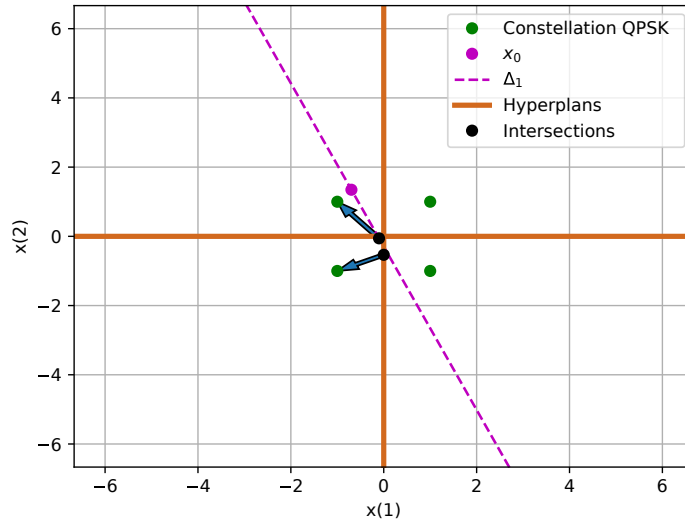


Figure 3.3) Sélection des points prometteurs comme les points de la constellation les plus proches des intersections  $\Delta_i \cap \mathcal{H}_i$ .

### 3.2.2 Recherche locale : exploitation

La recherche fine améliore la sélection précédente en examinant les solutions à proximité des points prometteurs. Cette étape accepte en entrée le sous-ensemble  $\mathcal{P}$  et produit la liste  $\mathcal{S}$  qui servira au calcul des LLR. Avant l'exploitation locale, l'ensemble  $\mathcal{S}$  est initialisé avec  $\mathcal{P}$ . Ensuite, un processus itératif est effectué pour chaque point prometteur de  $\mathcal{P}$ . Une itération est composée des trois étapes suivantes.

1. Générer au plus  $2N_t$  nouveaux points en faisant varier tour à tour chaque coordonnée du point initial. Pour chaque coordonnée modifiée, le symbole est remplacé par le précédent et le suivant dans la constellation à valeur réelle. Si le symbole à modifier est déjà à une valeur extrême de la constellation (*ie*, le premier ou le dernier symbole), seul un point est généré. Voici un exemple de ce procédé pour une constellation 16QAM à valeurs réelles comprenant les symboles  $\{-3, -1, 1, 3\}$  et pour le point de départ  $(1, 3)$  :

$$(1, 3) \dashrightarrow \begin{cases} (-1, 3) & \text{premier symbole précédent} \\ (3, 3) & \text{premier symbole suivant} \\ (1, 1) & \text{second symbole précédent} \\ & \text{pas de second symbole suivant} \end{cases} \quad (3.21)$$

Cet exemple produit donc trois nouveaux points.

2. Calculer la fonction objectif pour chaque nouveau point généré et l'ajouter à  $\mathcal{S}$ .
3. Sélectionner le meilleur point parmi le candidat initial et les nouveaux candidats générés. Le meilleur point devient le nouveau point de départ pour l'itération suivante. En cas d'égalité, le point initial est préféré.

Les trois étapes précédentes décrivent une recherche locale visant à améliorer un point de départ. La correction totale<sup>1</sup> de cette recherche est facile à prouver. En effet, l'algorithme est partiellement correct<sup>2</sup> puisque le point final de l'itération est toujours meilleur que le point initial ce qui en fait un invariant de boucle. De plus, l'algorithme termine<sup>3</sup> puisque la fonction objectif des points est une fonction décroissante sur un ensemble fini et puisque la règle d'égalité de l'étape 3 assure la terminaison dans le cas où deux points ont exactement la même valeur de fonction objectif.

Ainsi, la recherche locale est assurée d'améliorer le point initial et de terminer après un nombre fini d'itérations. Cependant, il est possible de fixer un nombre maximal d'itérations pour limiter la complexité de la détection, quitte à perdre en performance. Ce compromis sera étudié par une analyse de Pareto dans la suite du chapitre.

<sup>1</sup>En informatique, un algorithme est dit totalement correct s'il termine et qu'il est correct.

<sup>2</sup>En informatique, un algorithme est dit partiellement correct s'il fournit un résultat correspondant aux attentes. Ici, le point final doit être meilleur que tous ses voisins.

<sup>3</sup>En informatique, on dit qu'un algorithme termine lorsqu'il produit toujours un résultat en temps fini et on parle alors de sa terminaison. Un algorithme bouclant à l'infini est par exemple un algorithme qui ne termine pas.



#### 3.2.3 Positionnement du détecteur géométrique dans la taxonomie exploration/exploitation

Le chapitre 1 a amorcé une classification des détecteurs selon qu'ils explorent et/ou exploitent. La quasi-totalité des algorithmes de l'état de l'art exhibe une exploitation du modèle mais seule une petite partie d'entre eux explore. Ce chapitre avait également défini l'exploration par *la réalisation, à une étape donnée, de choix sous-optimaux dans l'espoir d'obtenir un meilleur résultat final*. Cette section vise à positionner le détecteur géométrique précédemment décrit dans cette taxonomie.

Comme tous les autres détecteurs de l'état de l'art, le détecteur géométrique est un détecteur à exploitation. En effet, la seule façon de ne pas exploiter du tout est de construire la liste  $\mathcal{S}$  à partir de points sélectionnés complètement au hasard. Dans notre cas, l'exploitation transparaît dans les deux étapes de l'algorithme. Tout d'abord, la recherche grossière exploite les informations du modèle à travers la SVD de la matrice du canal. De plus, la recherche fine est une exploitation pure puisqu'elle cherche à tirer le meilleur parti des points prometteurs qui lui sont fournis.

Tous les détecteurs à exploration rencontrés jusqu'ici implémentaient ce comportement par une approche aléatoire. C'est par exemple le cas des détecteurs bio-inspirés de la section 1.3.6 ainsi que de notre nouveau détecteur à parcours d'arborescence stochastique du chapitre 2. Cela dit, le détecteur géométrique est également un algorithme à exploration malgré l'absence de composantes aléatoires. En effet, il effectue bien une sélection de points sous-optimaux lors de la recherche grossière en espérant que la diversité des profils de  $\mathcal{P}$  produise un meilleur résultat final. A titre d'illustration, un détecteur géométrique sans exploration consisterait à construire  $\mathcal{P}$  avec les  $p$  meilleurs points sans obligation de diversité, c'est à dire en ne tenant pas compte du critère 2 de la section 3.2.1.

### 3.3 Atouts et faiblesses de la version historique

Cette section présente notre évaluation des qualités et faiblesses de ce détecteur géométrique sur le plan de la complexité, des performances et de l'implémentabilité telles que nous l'avons publiée dans [8]. Cette évaluation vise à mettre en lumière les principaux changements à apporter ainsi qu'à identifier les atouts à préserver sur lesquels capitaliser pour améliorer le détecteur historique.

#### 3.3.1 Analyse de la complexité arithmétique

L'analyse de complexité comprend deux aspects distincts : le pré-traitement effectué une fois par temps de cohérence du canal de transmission et la détection à proprement parler. Comme plusieurs centaines de vecteurs sont transmis par temps de cohérence, le pré-traitement n'impacte que faiblement la complexité une fois rapporté au nombre de vecteurs détectés. A titre de comparaison, la complexité du détecteur géométrique est ici rapportée à celle du détecteur k-best décrit en section 2.4.1

Commençons par comparer les pré-traitements. Le détecteur géométrique présenté

### 3 Diagnostic du détecteur géométrique historique

requiert une SVD et le calcul d'une pseudo-inverse par temps de cohérence. La section 3.1.2 a établi que la complexité arithmétique de la décomposition croît cubiquement avec la taille du système. De plus, nous avons démontré que deux produits matriciels et des inversions scalaires suffisent à obtenir la pseudo-inverse à partir de la SVD. Le pré-traitement du détecteur géométrique dans son ensemble évolue donc cubiquement avec la taille du système. Le détecteur k-best requiert quant à lui une décomposition QR qui s'obtient aussi par un algorithme de complexité cubique. Ainsi, les deux détecteurs possèdent des pré-traitements asymptotiquement équivalents que nous négligerons dans la suite de l'analyse devant les calculs effectués pour chaque vecteur.

Le tableau 3.1 détaille la complexité arithmétique du détecteur géométrique tel qu'il a été présenté précédemment. La complexité asymptotique en  $\Theta(N_t^2 N_r)$  est supérieure à celle du k-best qui a été évaluée à  $\Theta(N^2)$  dans le tableau 2.2. De plus, il est intéressant de noter que l'étape la plus impactante est l'évaluation de la fonction objectif pour chaque intersection. En effet, la recherche grossière produit  $DN_t$  intersections et chacune d'entre elle requiert un calcul de norme demandant  $N_t N_r$  produits-accumulations.

Pour autant, la taille du système MIMO n'est pas toujours suffisamment grande pour atteindre le comportement asymptotique. C'est pourquoi, la figure 3.4 compare la complexité arithmétique exacte en fonction du nombre d'antenne dans un système carré ( $N_t = N_r = N$ ). La similitude du nombre de produits et du nombre d'additions s'explique par l'impact prépondérant des évaluations des fonctions objectifs sur la complexité totale. De plus, les pentes confirment les complexités asymptotiques prévues par les deux tableaux. Enfin, le graphique montre que le détecteur géométrique est plus simple tant que la taille du système à valeur réelle est inférieure à 16, ce qui revient à 8 antennes physiques. Ainsi, ce détecteur est donc préférable pour ce critère pour les systèmes MIMO conventionnels tels que la WiFi ou le WiMAX.

#### 3.3.2 Performances de détection

Les performances de détection sont évaluées en termes de BER en fonction du SNR selon la méthode de Monte Carlo présentée en section 2.1.1. Le détecteur K-best est une nouvelle fois choisi comme référence et nous fixons  $K = 256$  pour maximiser ses performances dans le compromis performance-complexité. Le test est réalisé dans des conditions proches de celles décrites dans le standard WiFi : un canal 4x4 est utilisé conjointement à des QPSK ou 16-QAM. Selon la modulation, le cas testé correspondra à un modulation and coding scheme (MCS) de 1 ou de 3 selon le tableau 1.1 et le standard WiFi.

Le caractère prospectif de la comparaison nous pousse à choisir un codage de canal simple et nous retenons donc le code convolutif généré par  $(7, 5)_8$  qui a été introduit en section 1.2.4. Pour rappel, il s'agit d'une version simplifiée du code présent dans le standard WiFi qui est aussi un code convolutif de rendement 1/2. Un décodeur de Viterbi est utilisé à la réception avec une profondeur de 10 symboles, ce qui correspond aux ordres de grandeurs attendus pour un code convolutif de cette taille [41].

La figure 3.5 présente l'évolution du BER en fonction du SNR dans le cas d'un MCS de 1 (*ie*, avec une QPSK) pour les deux détecteurs considérés. Le SNR est défini

### 3.3 Atouts et faiblesses de la version historique

Étape	Nb $\times$	Nb $+$
Calculer $\mathbf{x}_0$	$N_t N_r$	$N_t N_r$
Intersections & Projections	$DN_t^2$	$DN_t^2$
Évaluation de la fonction objectif	$DN_t^2 N_r$	$DN_t^2 (N_r + 1)$
Une itération d'exploitation	$CDN_t(N_r + N_t)$	$CDN_t(N_r + N_t + 1)$
Calcul des LLR	0	1
<b>Complexité asymptotique</b>	$\Theta(N_t^2 N_r)$	$\Theta(N_t^2 N_r)$

Tableau 3.1) Complexité arithmétique du détecteur géométrique historique.

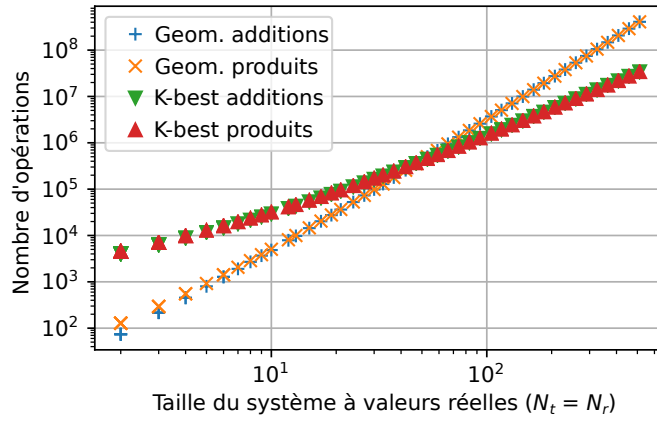


Figure 3.4) Comparaison de la complexité arithmétique du détecteur géométrique historique et du k-best ( $K = 252$ ) de [3].

### 3 Diagnostic du détecteur géométrique historique

selon (2.2). Nous remarquons que le détecteur géométrique fournit des performances équivalentes à la référence de la littérature jusqu'à un taux d'erreur de  $10^{-4}$ . Cependant, il requiert un environ 2dB de plus que le k-best pour atteindre le seuil des  $10^{-5}$ . Enfin, on note l'apparition d'un palier puisque le BER tend vers  $10^{-6}$  plutôt que de continuer à décroître.

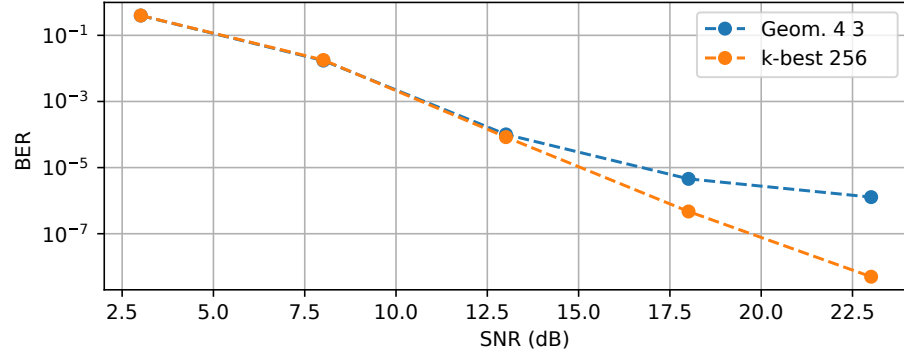


Figure 3.5) Comparaison des performances du détecteur géométrique historique et du k-best de [3] pour une QPSK et un codage convolutif de rendement 1/2. Cas proche du standard WiFi avec un MCS de 1.

Ces performances sont encourageantes mais montrent une limite du détecteur géométrique historique. En effet, les performances sont acceptables jusqu'aux BER de l'ordre de  $10^{-5}$ . De plus, l'analyse de complexité précédente a révélé que le détecteur géométrique était 6 fois plus simple que le k-best. Ainsi, le choix entre les deux détecteurs peut revenir à un choix entre de meilleures performances de détection mais coûteuses ou des performances acceptables mais moins coûteuses. Pour autant, la saturation du BER est plus problématique car certaines transmissions codées visent des taux d'erreur inférieurs à  $10^{-6}$ .

Les simulations avec une 16-QAM montrent que le détecteur géométrique ne parvient pas du tout à atteindre des performances acceptables. En effet, nous n'avons jamais observé de BER inférieur à  $10^{-2}$ , même pour un SNR supérieur à 25dB. La version historique du détecteur géométrique n'est donc pas du tout utilisable pour les modulations d'ordres supérieures.

#### 3.3.3 Implémentabilité sur architecture matérielle

L'implémentabilité du détecteur géométrique historique est évaluée à partir d'un prototype réalisé sur un système sur puce (system on chip, SoC) composé d'un FPGA 28nm et de deux co-processeurs ARM. Le FPGA est configuré grâce au Very high speed integrated circuit Hardware Description Language (VHDL) et le co-processeur ARM est programmé en langage C.

L'implémentation FPGA se concentre sur les étapes spécifiquement liées à la stratégie

### 3.3 Atouts et faiblesses de la version historique

de détection géométrique et délègue au co-processeur ARM les étapes qui sont déjà bien connues à savoir le calcul des LLR avec l'approximation max-log, la SVD et la pseudo-inverse. En effet, l'objectif de cette implémentation est d'évaluer les qualités des étapes spécifiquement liées à la stratégie de détection géométrique. De même, l'évaluation de l'implémentabilité est préférée au détriment de la recherche de performances. Ainsi, l'implémentation est construite comme un pipeline piloté par les données où chaque module est indépendant des autres ce qui permet de faciliter l'analyse. Le fonctionnement de ce pipeline est détaillé un peu plus tard.

Les dimensions du système MIMO sont fixées à  $N_t = N_r = N = 10$  pour simuler un cas très légèrement plus exigeant que les cas d'utilisation identifiés en partie 1.2.1. De plus, le prototype est limité à la détection des QPSK qui sont les seules modulations pour lesquelles les performances sont acceptables. Les nombres sont représentés dans un formalisme de virgule fixe et le nombre de bit alloués à chaque entrée est donné dans le tableau 3.2. La taille de la partie entière est choisie pour éviter les overflows dans 99% des cas rencontrés en simulation. Les variables intermédiaires et de sorties sont quantifiées de façon à éviter toute pertes de données pour échapper aux erreurs d'arrondis, quitte à augmenter l'utilisation des ressources.

	Partie entière	Partie fractionnaire	Longueur totale
$\mathbf{x}_0$	3 bits	3 bits	6 bits
$\{\mathbf{v}_i\}_{i=1}^D$	1 bit	5 bits	6 bits
$\mathbf{y}$	6 bits	2 bits	8 bits
$\mathbf{H}$	6 bits	2 bits	8 bits

Tableau 3.2) Quantification des entrées du détecteur.

### Architecture de l'implémentation matérielle

La figure 3.6 présente l'architecture générale sous la forme d'un flux de données mettant en évidence les modules. Les signaux de synchronisation ne sont pas représentés par soucis de lisibilité. Le contenu des modules dont le nom n'est pas suffisamment précis sera détaillé dans la suite.

L'architecture générale met en lumière que le détecteur géométrique est à la fois parallélisable et pipelinisable. En effet, trois niveaux de parallélisation sont disponibles. Tout d'abord, les macro-modules d'exploration (*ie* recherche grossière) et d'exploitation (*ie* recherche fine) peuvent être exécutés simultanément. Au sein de ces macro-modules, les modules sont également parallélisables puisque la projection, l'intersection et les calculs de normes sont par exemple réalisables en même temps. Enfin, chaque module donne lieu à une parallélisation interne classique : il est par exemple possible de traiter simultanément chaque coefficient d'un vecteur lors du calcul de sa norme. De plus, l'architecture générale se comporte comme un pipeline de profondeur 7 ce qui augmente significativement le débit. Ce dernier point exploite le déterminisme du détecteur géométrique qui n'effectue pas de retour arrière en fonction des entrées.

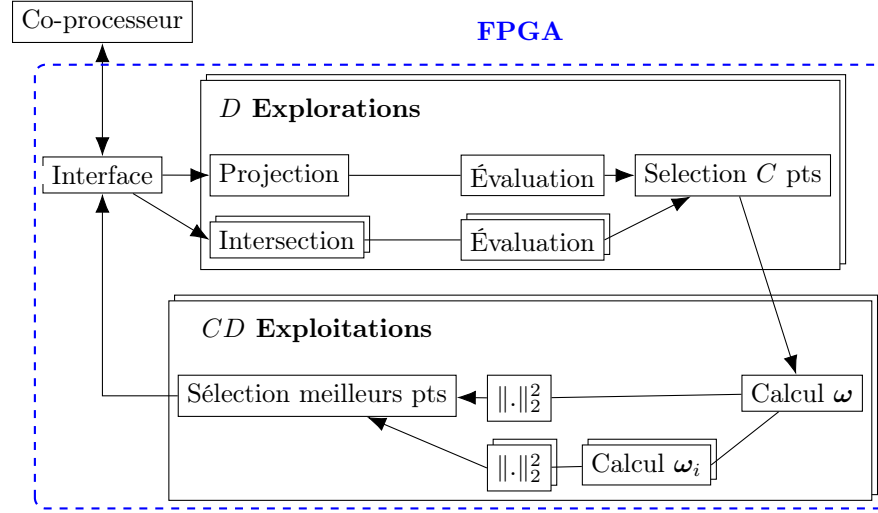


Figure 3.6) Architecture de l'implémentation sur FPGA sous une forme simplifiée.

### Description des modules

L'exploration géométrique est implémentée de la façon la plus directe possible. Ainsi,  $\mathbf{x}_0$  et tous les  $\Delta_i \cap \mathcal{H}_j$  sont calculés puis projetés sur la constellation par les modules *Projection* et *Intersection*. Le module *Évaluation* calcule ensuite la fonction objectif avant que les  $C$  meilleurs points ne soient sélectionnés.

Le macro-module d'exploitation introduit quant à lui une variable intermédiaire

$$\boldsymbol{\omega} = \mathbf{y} - \mathbf{H}\mathbf{x} \quad (3.22)$$

pour réduire la complexité sans perte de performance. Ainsi, une fois que  $\boldsymbol{\omega}$  est calculé une première fois, le changement de la  $i^{\text{ème}}$  coordonnée requis lors de la recherche locale peut s'écrire

$$\boldsymbol{\omega}_i = \mathbf{y} - \mathbf{H}\mathbf{x} = \boldsymbol{\omega} - \mathbf{H}\mathbf{z} \quad (3.23)$$

où toutes les composantes de  $\mathbf{z}$  sont nulles sauf la  $i^{\text{ème}}$ . De plus, pour une QPSK, la composante non nulle est  $z_i = -2x_i$  ce qui revient à calculer  $\boldsymbol{\omega}_i$  avec  $N$  décallages à gauche et  $N$  additions.

Deux solutions différentes ont été retenues pour l'implémentation des sélections selon qu'elles recherchent la meilleure valeur ou les  $C$  meilleures. Le premier cas peut être implémenté par un simple arbre de réduction où les valeurs d'entrées sont comparées deux à deux, étage après étage, jusqu'à ce qu'il ne reste que la meilleure. Cependant, ce processus ne peut fonctionner que lorsque l'on recherche la meilleure fonction objectif. Ainsi, cette méthode n'est pas adaptée pour la recherche des  $C$  meilleures fonctions objectifs. Le second cas de figure est implémenté avec  $C$  registres stockant les meilleures fonctions objectifs rencontrées jusqu'à présent. Ces registres sont initialisés avec la valeur maximale possible puis chaque candidat est inséré pour maintenir un

### 3.3 Atouts et faiblesses de la version historique

ordre croissant parmi les registres. Une fois toutes les entrées insérées, les registres contiennent les  $C$  meilleures fonctions objectifs. Le tableau 3.3 illustre cette stratégie pour sélectionner 3 candidats parmi 5. Cette méthode permet de traiter une entrée en  $C$  comparaisons et au plus  $C$  décalages, ce qui représente un coût matériel raisonnable.

Entrée	Insérée dans	Reg 0	Reg 1	Reg 2
26	Reg 0	max	max	max
15	Reg 0	26	max	max
25	Reg 1	15	26	max
19	Reg 1	15	25	26
34	Rejeté	15	19	25
<b>Résultat</b>		<b>15</b>	<b>19</b>	<b>25</b>

Tableau 3.3) Processus de sélection des  $C = 3$  meilleures valeurs de fonction objectif parmi 5 candidats.

#### Pipeline piloté par les données

Un pipeline piloté par les données est adapté aux prototypes car il permet de diviser le système en petits éléments indépendants tout en conservant une implémentation fonctionnelle. Ainsi, chaque partie élémentaire peut être étudiée sans subir l'influence du reste du système. Chaque module de la figure 3.6 possède donc des signaux de contrôle en plus des flux de données représentés pour permettre leur synchronisation.

Notre pipeline est régi par un protocole monorail biphasé tel qu'il est décrit dans [80]. Ainsi, un signal de requête et un acquittement sont ajoutés en plus des bus de données. Chaque module possède donc un port de requête  $R$  piloté par le module émetteur, un port d'acquittement  $A$  piloté par le module récepteur et un bus de données. La poignée de main synchronisant les modules suit le protocole suivant.

1. Initialement,  $R = A$  et l'état du bus de données n'est pas spécifié.
2. Lorsque le module émetteur a des données à transmettre, il met à jour le bus de données  $D$  à la valeur souhaitée et passe  $R$  à l'état de  $\bar{A}$ . Le bus de données doit désormais conserver son état.
3. Le module de réception remarque l'événement en testant  $R \oplus A$ , stocke les données de  $D$  et acquitte en passant  $A$  à l'état de  $R$ . Ce module peut commencer ses calculs.
4. Le module remarque l'acquittement en surveillant  $R \oplus A$ . Il peut désormais modifier l'état du bus de données s'il le souhaite.

Une attention supplémentaire est nécessaire lorsqu'il y a plusieurs émetteurs et/ou récepteurs. En effet, nous supposons que chaque module est indépendant, ce qui implique qu'il n'y a aucune garantie qu'ils soient tous prêt à envoyer ou recevoir des données exactement au même instant. Pour palier ce problème, nous utilisons des

*C-elements* de Muller, qui sont des portes logiques répliquant l'état de ses entrées à sa sortie si elles sont toutes identiques et qui conserve son état sinon [81]. Ainsi, l'insertion de *C-elements* sur les connexions *A* et/ou *R* permet de ne transmettre les poignées de main que lorsque tous les modules sont au même état.

## 3.4 Synthèse sur la détection géométrique historique : une stratégie prometteuse et des défauts à améliorer

Ce chapitre a introduit un détecteur construit autour du compromis exploration-exploitation à travers deux étapes : une recherche globale utilisant des considérations géométriques suivie d'une exploitation grâce à une recherche fine. Un diagnostic complet de la stratégie géométrique a été mené afin d'en saisir les aspects mathématiques, algorithmique et d'implémentation ainsi que pour en évaluer les atouts et défauts.

Ainsi, les simulations de Monte Carlo ont montrées l'intérêt de l'approche géométrique lorsque la transmission utilise une QPSK et un nombre assez réduit d'antennes. Le détecteur étudié s'est montré plus simple qu'un détecteur à parcours d'arborescence tout en approchant leurs performances pour les SNRs faibles à moyens. De plus, un prototype sur SoC a mis en lumière des caractéristiques très propices à l'implémentation matérielle : forte parallélisation possible et comportement sans branches conditionnelles facilitant l'utilisation de pipelines.

Pour autant, plusieurs défauts ont été relevés que les travaux présentés dans le prochain chapitre chercheront à résoudre. Ainsi, un palier prévient la décroissance du BER à fort SNR dans le cas des transmissions en QPSK. De plus, le détecteur géométrique historique ne parvient pas à fournir des BER acceptables pour les modulations d'ordre supérieur qui sont pourtant présentes dans les standards retenus dans nos cas d'utilisation (Cf. section 1.2.1). Enfin, une amélioration de la complexité asymptotique est requise pour généraliser l'utilisation de la stratégie géométrique à des systèmes MIMO de taille plus conséquente.



## 4 Améliorations, études et implémentation de la détection géométrique

Le chapitre précédent a introduit la détection géométrique comme un algorithme intégrant le compromis exploration-exploitation au sein de sa structure. De plus, les études préliminaires ont révélé un potentiel tant en termes de performance de détection et de complexité qu'en termes d'implémentabilité. Cependant, un certain nombre de limites ont également été soulevées. Tout d'abord, la figure 3.5 montre que le détecteur géométrique ne maintient pas la décroissance du BER pour une QPSK à partir d'un certain SNR. De plus, le détecteur historique présenté est incapable de détecter les données émises avec des modulations d'ordre supérieur. Enfin, la complexité arithmétique du détecteur croît trop rapidement avec le nombre d'antennes d'émission et de réception du système. En effet, la version historique est plus simple pour les systèmes carrés à moins de 8 antennes mais sa croissance cubique le rend trop complexe pour les systèmes de taille moyenne à grande.

Ce chapitre propose donc plusieurs améliorations algorithmiques pour résoudre les défauts soulevés dans le chapitre précédent. Ces améliorations, détaillées dans la section 4.1, portent majoritairement sur la recherche globale et sur des optimisations de complexité. Toutes ces améliorations sont évaluées dans la section 4.2 afin de s'assurer qu'elles ont bien résolues les problèmes soulevés dans la version historique. Enfin, la section 4.3 présente une implémentation matérielle sur FPGA et compare les résultats obtenus avec des implémentations de référence issues de la littérature.

### 4.1 Améliorations algorithmiques de la détection géométrique

Nous proposons ici une série d'améliorations à la détection géométrique dont la majorité ont été publiées dans [9]. Tous les changements de cette section se fixent comme objectif d'améliorer, ou *a minima* de ne pas dégrader, la complexité arithmétique.

Il est possible de classer les améliorations présentées en trois catégories : les changements de la recherche locale (section 4.1.1), les changements de la recherche globale (sections 4.1.2 et 4.1.3) et les améliorations de complexité à l'échelle de l'ensemble du détecteur (sections 4.1.4 et 4.1.5).

### 4.1.1 Majoration du nombre d'itérations

La version originale de la recherche locale effectuait des itérations jusqu'à atteindre un point stable. La section 3.2.2 a démontré la terminaison de l'algorithme local mais aucune étude sur le nombre d'itérations requis pour atteindre le point stable n'a été proposée pour le moment. Or, la recherche locale est la principale étape dont le temps d'exécution dépend des entrées et il est connu que les comportements conditionnels produisent toujours des implémentations matérielles moins efficaces. C'est pourquoi, nous explorons ici la possibilité de majorer ou de fixer le nombre d'itérations plutôt que de laisser la recherche locale atteindre un point stable. La sortie de l'algorithme serait alors le résultat après un nombre pré-défini d'itérations quel que soit l'état de la convergence.

Le tableau 4.1 présente la probabilité que le point stable soit atteint après un certain nombre d'itérations en fonction du SNR. L'étude est réalisée pour un système MIMO  $10 \times 10$  à détection dure avec  $D = 2$  directions singulières et  $C = 2$  candidats retenus par direction. La plage de SNR balaye les points de fonctionnement depuis 0dB jusqu'à un SNR permettant de passer le BER sous les  $10^{-3}$  dans la configuration proposée. Nous notons, tout d'abord, que le SNR influe sur les probabilités puisqu'un peu plus d'itérations sont nécessaires quand le canal est particulièrement bruité. Ce constat correspond aux attentes puisqu'un fort bruit implique une recherche globale (*ie*, l'exploration) plus approximative et donc plus de travail pour la recherche locale (*ie*, l'exploitation). De plus, une itération est suffisante dans plus de 60% des cas et que 2 conviennent dans plus de 85% des cas quel que soit le niveau de bruit.

Nb its	0dB	2dB	4dB	6dB	8dB	10dB	12dB	14dB	16dB	18dB
1	61	68	72	73	77	80	82	84	85	86
2	27	24	23	22	19	17	15	13	13	12
3	9	6	4	5	3	3	2	2	2	2
4	3	2	1	1	1	1	1	0	0	0
5+	0	0	0	0	0	0	0	0	0	0

Tableau 4.1) Probabilités (en pourcents) du nombre d'itérations de recherche locale avant d'atteindre un point stable en fonction du SNR ( $C = 2$ ,  $D = 2$ , canal  $5 \times 5$ , sorties dures).

Les résultats précédents motivent la majoration du nombre d'itérations de recherche locale. En effet, le tableau 4.1 suggère qu'interdire la recherche de faire plus d'une ou deux itérations pourrait avoir un impact assez faible sur le déroulement de l'algorithme tout en rendant l'exécution plus prévisible. Ainsi, l'implémentation matérielle serait plus efficace tout en conservant des performances très proches. La figure 4.1 évalue les performances de détection dans la même configuration pour un nombre d'itérations fixé. La courbe *inf* représente le cas d'une majoration infinie, c'est à dire une recherche locale qui attend aussi longtemps que nécessaire pour atteindre le point stable. Le BER à nombre d'itérations fixé est nécessairement moins bon que celui sans limite d'itérations puisque la majoration impose d'utiliser un résultat intermédiaire plutôt

que final. Ainsi, le BER peut, au mieux, être équivalent si le nombre d'itérations requis est toujours plus grand que la limite imposée.

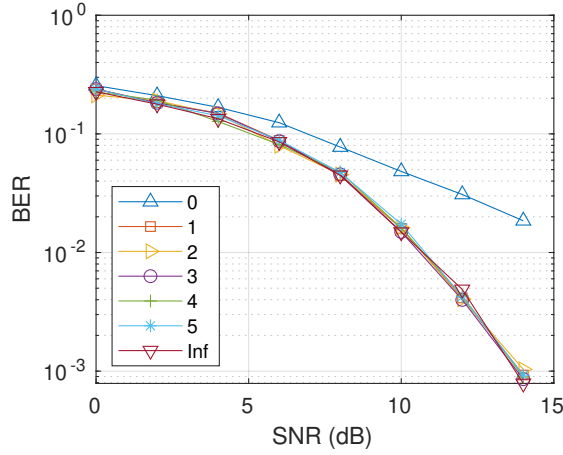


Figure 4.1) Évolution du BER en fonction du nombre maximum d'itérations de recherche locale ( $C = 2$ ,  $D = 2$ , canal  $10 \times 10$ , sorties dures).

Sans surprise, passer de 0 à 1 itération améliore significativement le résultat puisque sans recherche locale, le résultat ne dépend que de la recherche globale qui n'est pas construite pour cette tâche. Cependant, ajouter plus d'itérations n'augmente plus les performances ce qui suggère qu'une implémentation avec une seule itération fournit des BER équivalents à une implémentation avec exécution conditionnelle tout en étant plus efficace.

Les résultats précédents ont été obtenus pour une configuration particulière de taille de canal, de paramètre de détecteur et de type de sorties. Ainsi, rien ne garantit que le fait qu'une seule itération soit suffisante se transpose directement aux autres configurations. Pour autant, cette étude préliminaire suggère que la majoration du nombre d'itérations est une piste à explorer pour simplifier l'implémentation sans détériorer les performances. C'est pourquoi, les analyses de Pareto de la section 4.2 qui visent à déterminer les réglages optimaux incluront le majorant du nombre d'itérations comme paramètre du détecteur.

##### 4.1.2 Projection sur les sous-constellations et sorties souples

D'après la section 3.2, le détecteur géométrique s'appuie sur l'approximation max-log avec une liste définie par (1.25) pour produire des sorties souples. Or, la version avec liste donne le même résultat que la version exacte (1.24) si et seulement si la liste  $\mathcal{S}$  contient les deux minima que nous appelons minimum global et meilleure contre-hypothèse. L'algorithme de détection historique ne garantit en aucune façon la présence de ces deux minima. Pire encore, rien n'est fait pour rechercher spécifique-

ment les contre-hypothèses puisque le procédé actuel se concentre spécifiquement sur les candidats dans les meilleures zones de la fonction objectif. En effet, les contre-hypothèses nécessaires pour calculer (1.24) ne se trouvent pas toujours dans les zones de faible fonction objectif. Si c'était le cas, alors les LLR seraient systématiquement proches de zéro d'après (1.24), ce qui n'est pas le cas. C'est pour ces raisons que nous proposons une modification de la recherche globale afin d'inclure les contre-hypothèses. De plus, nous nous imposons de faire cette modification sans augmenter la complexité asymptotique pour ne pas aller à l'encontre de l'objectif général fixé dans l'introduction de cette section.

### Principe de la projection sur les sous-constellations

Pour rappel, la recherche globale construit les candidats en deux étapes : d'abord une construction géométrique de points puis une projection de ces points sur la constellation  $\mathcal{Q}^{N_t}$ . La façon la plus directe de former des points dans les

$$\chi_i^k = \{x \in \mathcal{Q}^{N_t} : b_i = k\} \quad (4.1)$$

est de projeter sur chacun de ces sous-ensembles plutôt que sur  $\mathcal{Q}^{N_t}$ . Bien qu'efficace, cette méthode comporte un sur-coût très important lié au nombre de sous-ensembles. En effet, il y a deux sous-ensembles par bit  $b_i$  ce qui implique que cette méthode requiert  $2qN_t$  projections avec  $q = \text{card}(\mathcal{Q})$ . Cette méthode n'est donc pas acceptable puisque sa complexité croît avec le nombre d'antennes.

Nous proposons donc une nouvelle méthode de projection basée sur la démodulation à valeur complexe. Soit  $f$  la fonction de démodulation de la constellation complexe définie par

$$f : \begin{cases} \mathcal{Q}^{\mathfrak{S}} \longrightarrow \{0, 1\}^{\log_2 q^{\mathfrak{S}}} \\ s \longmapsto (b_1, b_2, \dots, b_{\log_2 q}) \end{cases} \quad (4.2)$$

qui associe à chaque symbole le mot binaire correspondant. Nous pouvons maintenant définir les sous-constellations complexes

$$\mathcal{Q}_{i,k}^{\mathfrak{S}} = \{s \in \mathcal{Q}^{\mathfrak{S}} : f(s)_i = k\} \quad (4.3)$$

qui regroupent tous les symboles dont le  $i^{\text{ème}}$  bit associé est fixé à  $k$ . Ces ensembles diffèrent des  $\chi_i^k$  définis dans (4.1) par le fait qu'ils sont un sous-ensemble de la constellation (*ie*, il contient des symboles) plutôt que de  $\mathcal{Q}^{N_t}$  (*ie*, il contient des vecteurs de symboles).

Avec ces définitions, nous proposons maintenant de projeter les candidats construits par la méthode géométrique sur toutes les sous-constellations  $\mathcal{Q}_{i,k}^{\mathfrak{S}}$ . Contrairement à la projection sur les  $\chi_i^k$ , cette méthode ne requiert que  $2q^{\mathfrak{S}} = 4q$  projections, ce qui reste indépendant de la taille du système. De plus, elle assure que la liste des candidats finale contienne au moins un candidat dans chaque  $\chi_i^k$ .

Il est nécessaire de faire la projection sur les constellations complexes pour avoir assez de symboles différents. En effet, dans le cas d'une QPSK, la constellation réelle ne contient que deux symboles : -1 et 1. Il n'est alors pas possible d'extraire des

#### 4.1 Améliorations algorithmiques de la détection géométrique

sous-constellations de taille plus réduite et la méthode proposée est donc inutile. A l'inverse, la constellation complexe

$$\mathcal{Q}^{\mathbb{S}} = \{1 + j, 1 - j, -1 - j, -1 + j\} \quad (4.4)$$

offre quatre sous-constellations. Le tableau 4.2 synthétise le nom et la composition des quatre sous-constellations complexes et donne le processus équivalent sur les variables à valeurs réelles que nous utilisons.

Nom	Sous-constellation complexe	Projection équivalente réelle
$\mathcal{Q}_{1,1}$	$\{1 + j, 1 - j\}$	$N_t/2$ premiers bits à 1
$\mathcal{Q}_{2,1}$	$\{1 + j, -1 + j\}$	$N_t/2$ derniers bits à 1
$\mathcal{Q}_{1,0}$	$\{-1 + j, -1 - j\}$	$N_t/2$ premiers bits à 0
$\mathcal{Q}_{2,0}$	$\{1 - j, -1 - j\}$	$N_t/2$ derniers bits à 0

Tableau 4.2) Projection sur les sous-constellations en QPSK.

##### 4.1.3 Retirer les intersections pour réduire la complexité

L'évaluation de la fonction objectif de toutes les intersections est la seule opération responsable de la complexité asymptotique cubique d'après le tableau 3.1. En effet, la recherche globale calcule l'intersection de  $D$  droites avec  $N_t$  hyperplans, ce qui produit  $DN_t$  intersections. Pour chacune de ces intersections, un calcul de la fonction objectif requiert une multiplication  $\mathbf{H}\mathbf{x}$  dont la complexité évolue en  $N_t N_r$ . Ainsi, l'évaluation des  $DN_t$  intersections évolue bien en  $\Theta(N_t^2 N_r)$ , soit une complexité cubique. C'est pourquoi, nous proposons une nouvelle technique pour construire les candidats à complexité quadratique ce qui alignerait le détecteur géométrique avec le reste de la littérature (Cf. tableaux 2.2 et 2.3).

D'après l'analyse précédente, une recherche globale quadratique requiert de construire un nombre de candidats indépendant de la taille du système. En effet, le nécessaire calcul du produit  $\mathbf{H}\mathbf{x}$  est déjà quadratique. La présentation de la recherche globale dans la section 3.2.1 donnait les trois objectifs poursuivis pour construire la liste de candidats  $\mathcal{P}$  :

1. les points de  $\mathcal{P}$  doivent être prometteurs, c'est-à-dire avoir une fonction objectif faible ;
2. les points de  $\mathcal{P}$  doivent être aussi différents que possible ;
3. le cardinal  $p \triangleq \text{card}(\mathcal{P})$  doit être le plus faible possible.

Le troisième critère est désormais modifié pour devenir *le cardinal  $p \triangleq \text{card}(\mathcal{P})$  est indépendant de la taille du système* afin d'obtenir la complexité quadratique souhaitée.

Un premier point prometteur est évidemment  $\mathbf{x}_0$  défini par (1.11) puisqu'il correspond au minimum global de la fonction objectif sur  $\mathbb{R}^{N_t}$ . L'objectif est donc de construire des points à la fois très différents et suffisamment prometteurs à partir de

ce point de départ. Pour ce faire, notons que l'équation (3.16) indique que se déplacer selon les premières directions propres impacte peu la fonction objectif. Ainsi, partir de  $\mathbf{x}_0$  et se déplacer selon ces directions est une solution efficace pour atteindre l'objectif fixé.

Soit  $n_d \leq N_t$  le nombre de directions singulières considérées. Les candidats de  $\mathcal{P}$  peuvent être construits par la projection sur  $\mathcal{Q}^{N_t}$  des  $2^{n_d}$  points

$$\mathbf{x}_0 + \sum_{i=1}^{n_d} \frac{\pm f}{\lambda_i} \mathbf{v}_i \quad (4.5)$$

avec  $f$  un paramètre d'échelle dont nous discuterons plus tard. Ces points vérifient bien les critères attendus. En effet, ils sont également prometteurs puisqu'ils ont tous la même fonction objectif qui est proche de celle de  $\mathbf{x}_0$ . De plus, ils sont les plus différents possibles pour des points appartenant à la même isovaleur puisque nous nous sommes déplacés dans le long des directions singulières les moins impactantes. Enfin, ils sont au nombre de  $2^{n_d}$  qui est bien indépendant de la taille du système.

La figure 4.2 illustre graphiquement le principe dans le cas d'une 16-QAM avec  $n_d = 1$  ou  $n_d = 2$ . On retrouve en bleu les deux vecteurs singuliers  $\mathbf{v}_1$  et  $\mathbf{v}_2$  et en violet la projection sur la constellation. Les isovaleurs sont également représentées pour illustrer que les points construits ont tous la même valeur de fonction objectif. Dans cet exemple, le cas  $n_d = 1$  produit  $\mathcal{P} = \{(-3, -1), (-1, 3), (-1, 1)\}$  en projetant les deux points construits ainsi que  $\mathbf{x}_0$ . De même, le cas  $n_d = 2$  produit  $\mathcal{P} = \{(-3, -1), (-3, 1), (-1, 3), (-1, 1)\}$ .

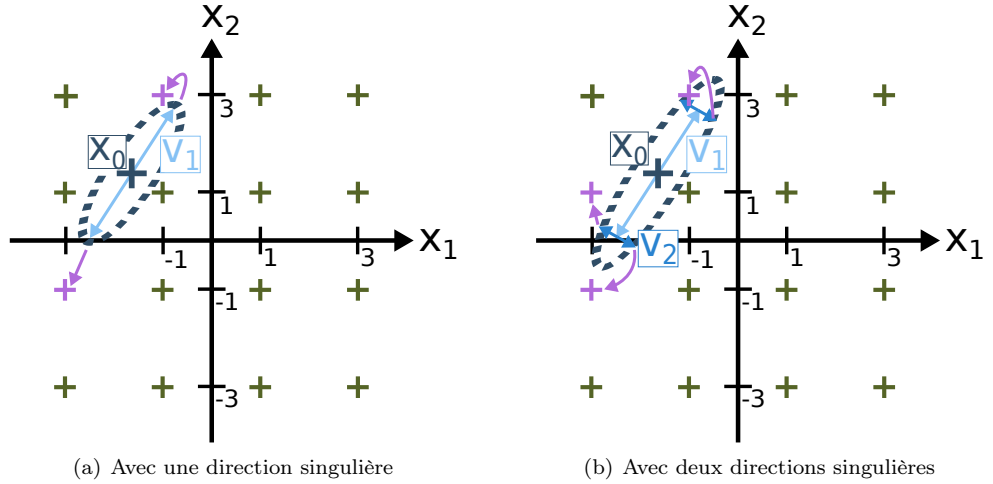


Figure 4.2) Principe de construction de candidats sans intersections pour un canal bien conditionné dans le cas d'une 16-QAM.

La figure 4.3 présente la même construction dans le cas d'un canal mal conditionné où l'ellipsoïde d'isovaleur perd une dimension. Il est évident que l'ensemble  $\mathcal{P} =$

$\{(-3, -3), (1, 3), (-1, 1)\}$  ne remplit plus les critères fixés en début de section. En effet, les points sont bien différents mais ils ne sont pas du tout prometteurs. La figure met en lumière que le problème vient des points de (4.5) qui sont en dehors de la constellation. La projection sur  $Q^{N_t}$  produit alors des résultats inadéquats. C'est pour résoudre ce cas de figure que le facteur multiplicateur  $f$  est ajouté à l'équation (4.5). En effet, en réduisant la norme du vecteur  $v_1$ , les points projetés ne sont plus hors de la constellation et les points remplissent de nouveau les conditions recherchées.

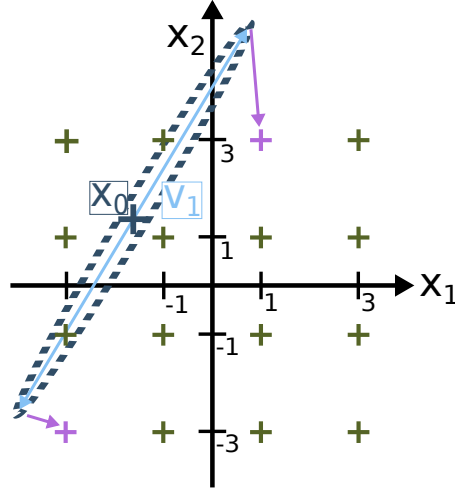


Figure 4.3) Principe de construction des candidats sans intersection pour un canal mal conditionné, une 16-QAM et  $n_d = 1$

La sélection du facteur  $f$  impacte donc significativement les résultats. En effet, si les vecteurs singuliers sont trop courts, tous les points sont proches les uns des autres et la recherche globale n'explore plus l'espace des possibles. A l'inverse, s'ils sont trop longs, les points produits ne sont plus prometteurs ce qui met en péril la recherche locale autour de ces mauvais points. Une décision sur la valeur de  $f$  peut être prise en fonction de la qualité du canal mais elle nécessite des calculs supplémentaires et donc un surcoût de complexité. C'est pourquoi, nous proposons de pré-sélectionner un petit nombre de  $f$  et de calculer (4.5) pour chacun d'eux. Ces facteurs doivent être assez différents pour assurer qu'au moins un jeu de  $\frac{f}{\lambda_i} \mathbf{v}_i$  soit de taille convenable. Dans ces conditions, le nombre de candidats construits est  $p \leq 1 + n_f n_d$ , avec  $n_f$  le nombre de facteurs, ce qui reste bien indépendant de la taille du système.

#### 4.1.4 Opérations au niveau bloc et au niveau vecteur

La section 1.2.2 a relevé que le canal restait stable pendant un laps de temps appelé *temps de cohérence* qui est suffisamment long pour transmettre plusieurs centaines de vecteurs [31]. Cette caractéristique permet de réduire significativement la complexité des détecteurs en séparant les calculs à effectuer une fois par temps de cohérence et

ceux à effectuer pour chaque vecteur reçu. Nous appellerons dans la suite du manuscrit *opérations au niveau bloc* l'ensemble des opérations qui ne sont effectuées qu'une fois par temps de cohérence. Les autres opérations seront nommées *opérations au niveau vecteur*. Cette section scinde toutes les opérations du nouveau détecteur géométrique entre ces deux catégories pour éviter les calculs redondants. Une opération peut être effectuée au niveau du bloc si, et seulement si, elle ne dépend que de l'état du canal et qu'elle est indépendante du vecteur reçu  $\mathbf{y}$ .

La suite de cette section présente les opérations niveau bloc et niveau vecteur et propose une synthèse graphique du détecteur à travers les figures 4.4 et 4.5.

### Opérations niveau bloc

Deux opérations présentées précédemment peuvent être calculées une fois par temps de cohérence. Tout d'abord, le calcul de la SVD et de la pseudo-inverse de Moore-Penrose décrits en section 3.1 sont évidemment indépendants de  $\mathbf{y}$ . Le pré-traitement au niveau bloc de ces calculs matriciels est d'autant plus important compte tenu de leur complexité. Ensuite, les  $\sum_{i=1}^{n_d} \frac{\pm f}{\lambda_i} \mathbf{v}_i$  de l'exploration géométrique introduits en section 4.1.3 sont également indépendants du vecteur  $\mathbf{y}$ . Ces  $2^{n_d}$  sommes peuvent donc être pré-calculées au niveau bloc.

Enfin, une dernière opération moins évidente peut être effectuée au niveau bloc. Pour rappel, les évaluations de fonctions objectifs requièrent de calculer des produits matriciels  $\mathbf{H}\mathbf{x}$  pour différents  $\mathbf{x} \in \mathcal{Q}^{N_t}$ . Ces produits peuvent être réexprimés en

$$\mathbf{H}\mathbf{x} = \sum_{j=1}^{N_t} \mathbf{h}_j x_j \quad (4.6)$$

avec  $\mathbf{h}_j$  la  $j^{\text{ème}}$  colonne de  $\mathbf{H}$ . Le calcul naïf de (4.6) requiert  $N_t$  produits pour chaque  $\mathbf{h}_j x_j$  puis  $N_t^2$  additions pour sommer tous les vecteurs. Cependant,  $x_j$  est à valeurs dans l'ensemble fini  $\mathcal{Q}$  de sorte qu'il n'y a que  $q$  produits  $\mathbf{h}_j x_j$  possibles. Par exemple, avec  $\mathcal{Q} = \{-3, -1, 1, 3\}$  (c'est-à-dire en utilisant une 16-QAM), un produit  $\mathbf{h}_j x_j$  ne peut être que  $-3\mathbf{h}_j$ ,  $-\mathbf{h}_j$ ,  $\mathbf{h}_j$  ou  $3\mathbf{h}_j$ . Cette propriété permet de pré-calculer et de stocker les  $q$  produits  $\mathbf{h}_j x_j$  possibles au niveau bloc. Cela équivaut, par exemple, à prétraiter et stocker les matrices  $-3\mathbf{H}$ ,  $-\mathbf{H}$ ,  $\mathbf{H}$  et  $3\mathbf{H}$  pour la 16-QAM et dans le cas général, à stocker et calculer l'ensemble  $\{s\mathbf{H} : s \in \mathcal{Q}\}$ . Ainsi, le calcul de (4.6) au niveau vecteur devient la somme de  $N_t$  vecteurs connus et ne nécessite aucun produit. Enfin, l'espace de stockage et le nombre de précalculs peuvent être réduits de moitié pour les constellations usuelles. En poursuivant l'exemple précédent de la 16-QAM, le stockage de  $\mathbf{H}$  et  $3\mathbf{H}$  est suffisant pour calculer (4.6) avec seulement des additions puisque  $-\mathbf{H}$  et  $-3\mathbf{H}$  peuvent être obtenus en soustrayant plutôt qu'en ajoutant les vecteurs prétraités.

La figure 4.4 synthétise graphiquement l'ensemble des opérations qu'il est possible d'effectuer au niveau bloc. Le pré-traitement de ces grandeurs réduit fortement la complexité globale du détecteur puisqu'elles ne sont évaluées qu'une fois par temps de cohérence.



### Opérations niveau vecteur

L'ensemble des opérations niveau bloc vient d'être présenté et toutes les autres tombent donc dans la catégorie des opérations niveau vecteur. La figure 4.5 résume graphiquement l'ensemble de ces opérations. On y retrouve le calcul de  $\mathbf{x}_0$  à partir du vecteur reçu ainsi que de la pseudo-inverse évaluée et stockée au niveau bloc. Ce vecteur est ensuite ajouté à chaque sommes pré-calculées et le tout est projeté sur les constellations et sous-constellations selon le processus décrit en section 4.1.2. Enfin, les itérations de recherches locales permettent d'enrichir la liste totale de points qui sert à calculer les LLR à partir de l'approximation max-log. Cette figure met également en avant les deux composantes de la détection géométrique : l'exploration géométrique de la recherche globale et l'exploitation à travers la recherche locale.

#### 4.1.5 Réutilisation des fonctions objectifs

L'étude de notre détecteur à parcours d'arborescence stochastique introduit en section 2.5.1 avait révélé que certains nœuds étaient visités plusieurs fois. Pour améliorer la complexité, nous avons alors proposé de stocker les fonctions objectifs pour ne pas les ré-évaluer plusieurs fois. Cette section propose de transposer cette amélioration à notre nouveau détecteur géométrique.

Dans le cas de notre détecteur géométrique, le risque de redondance apparaît lors de la constitution de l'ensemble  $\mathcal{P}$  par la recherche globale. En effet, la méthode décrite en section 4.1.3 peut ajouter de nombreuses fois le même point. L'exemple de la figure 4.2(b) génère par exemple deux fois le point  $(-1, 3)$ . Ce phénomène est d'autant plus présent lorsque l'on ajoute la projection sur les sous-constellations de la section 4.1.2.

La figure 4.6 quantifie ce phénomène pour une application typique issue des cas d'utilisation de la section 1.2.1. Les statistiques sont évaluées pour une communication WiMAX utilisant une QPSK et un code LDPC avec des blocs de 720 bits et un rendement de  $1/2$ . Ce type de configuration de codage et modulation adaptatifs (adaptive coding and modulation, AMC) correspond aux cas d'utilisation pour un canal très bruité et nous fixons donc le SNR à 10dB. La matrice de vérification du code LDPC est construite selon la norme WiMAX [29]. Les valeurs de  $f$  sont fixées à des valeurs typiques de réglages qui seront justifiées dans la suite du chapitre.

Les boîtes de Tukey relèvent d'abord que le nombre de points uniques diminue fortement lorsque l'on ajoute des dimensions. Ce résultat est attendu puisque l'ajout de dimension implique la création de points en plus grand nombre et augmente donc les chances d'avoir des projections identiques. La figure 4.2 est une bonne illustration de ce phénomène.

De plus, ces statistiques montrent une très forte redondance des points de  $\mathcal{P}$  avec 45% de points différents dans le pire cas et moins de 10% dans les meilleurs. Ainsi, il est intéressant d'éviter de calculer la fonction objectif plusieurs fois pour un même point. Une solution simple est de stocker les fonctions objectifs précédemment évaluées et de lire la valeur stockée plutôt que d'effectuer le calcul une seconde fois. Une meilleure solution est de purger  $\mathcal{P}$  des redondances avant de le transmettre à la suite

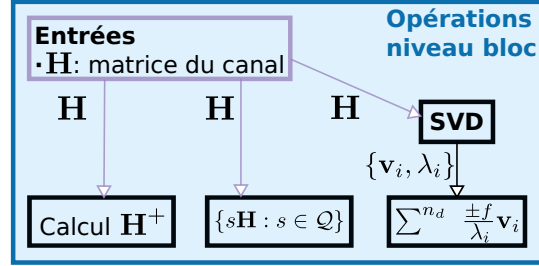


Figure 4.4) Synthèse graphique des opérations niveau bloc

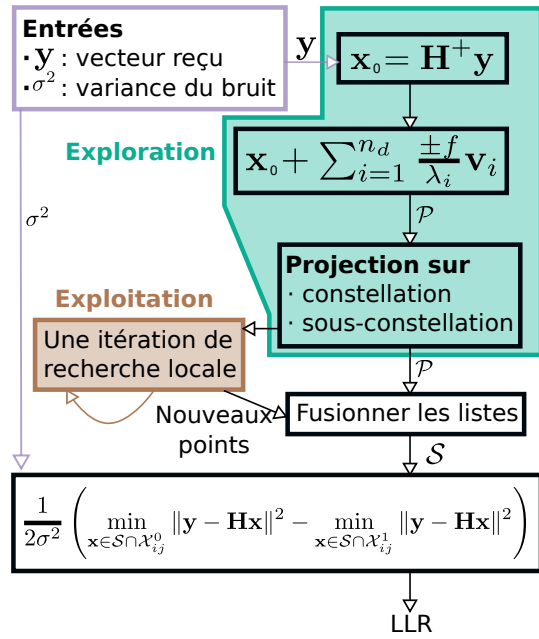


Figure 4.5) Synthèse graphique des opérations niveau vecteur

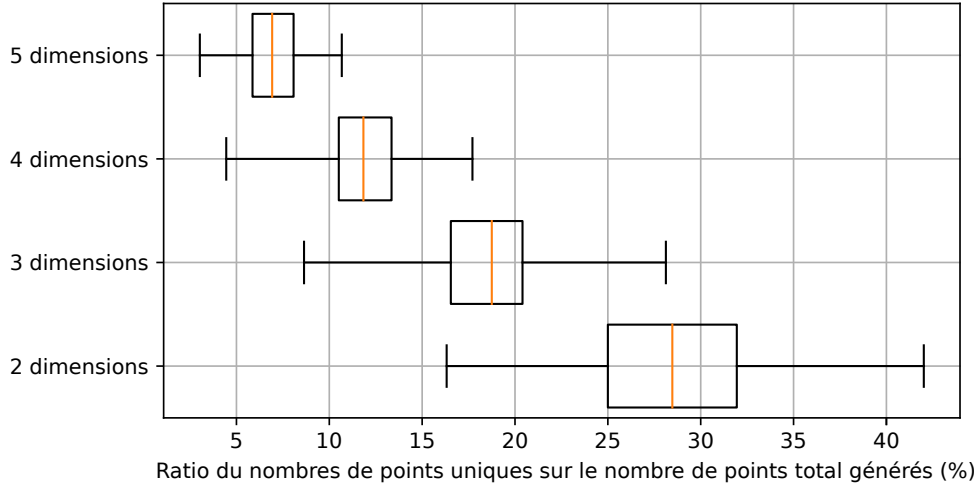


Figure 4.6) Statistiques des points générés en fonction de  $n_d$  avec  $f \in \{1/4, 4\}$ . Les moustaches représentent les valeurs minimales et maximales rencontrées.

du détecteur. En effet, cela permet de réduire le nombre de recherches locales et la taille de la liste  $\mathcal{S}$  utilisée pour l'approximation max-log.

## 4.2 Analyses de Pareto des améliorations

Cette section évalue la pertinence des améliorations proposées pour résoudre les problèmes soulevés dans le chapitre 3. Ainsi, nous souhaitons vérifier que la complexité asymptotique du détecteur est bien quadratique. De plus, nous attendons une augmentation des performances lors des transmissions utilisant une QPSK. Enfin, le nouveau détecteur géométrique doit être capable de détecter des messages modulés par une 16-QAM, ce que ne pouvait pas faire la version historique.

### 4.2.1 Complexité arithmétique asymptotique

Le tableau 4.3 quantifie la complexité de chaque étape du nouveau détecteur géométrique ainsi que sa complexité asymptotique. Seule la complexité des opérations niveau vecteur est prise en compte comme c'était le cas pour les autres détecteurs.

Nous supposons que toutes les améliorations présentées précédemment ont été utilisées. Ainsi, le nombre de fonctions objectifs à évaluer est  $p < 2^{n_d} n_f$  puisque l'on a retiré les redondances de l'ensemble  $\mathcal{P}$ . De plus, les calculs effectués au niveau bloc permettent une recherche globale sans produit, à l'exception des carrés pour la norme de la fonction objectif.

Le tableau confirme que le changement de méthode d'exploration a permis d'atteindre une complexité quadratique. Ainsi, la complexité du nouveau détecteur croît de la

même façon que les autres références (voir par exemple [2–4]), ce qui améliore son application aux systèmes de grandes tailles.

Étape	Nb $\times$	Nb $+$
Calcul de $\mathbf{x}_0$	$N_t N_r$	$N_t N_r$
Addition des sommes selon (4.5)	0	$2^{n_d} n_f N_t$
Évaluation de la fonction objectif	$p N_t$	$2p N_t$
Une itération d’exploitation	$p N_t (N_r + N_t)$	$p N_t (N_r + N_t + 1)$
Calcul des LLR	0	1
<b>Complexité asymptotique</b>	$\Theta(N_t^2 + N_t N_r)$	$\Theta(N_t^2 + N_t N_r)$

Tableau 4.3) Complexité arithmétique du nouveau détecteur géométrique.

### 4.2.2 Analyses de Pareto

Cette section présente les configurations testées et analyse les résultats obtenus dans le paradigme de Pareto présenté en section 2.1.3.

#### Configuration des simulations

Deux modulations différentes sont testées (QPSK et 16-QAM) en accord avec les objectifs à valider. Pour chacune d’entre elles, une transmission codée correspondant au standard WiMAX est simulée. Ainsi, l’émetteur code le message binaire avec le code LDPC de taille de bloc 720 bits et de ratio 1/2 construit selon la norme WiMAX [29]. Après transmission dans un canal 4x4, le signal est détecté puis décodé par 15 itérations de détection par propagation des convictions (belief propagation, BP) selon l’algorithme min-sum.

Le tableau 4.4 énumère tous les paramètres testés pour chaque détecteur évalué. Les valeurs déjà utilisées pour les deux références de la littérature et pour le parcours d’arbre stochastique dans la section 2.5 sont reconduits pour les mêmes raisons.

Quatre paramètres sont considérés dans le cas du nouveau détecteur géométrique : la projection sur les sous-constellations ou non (Cf. section 4.1.2), le nombre maximum d’itérations autorisées pour la recherche locale (Cf. section 4.1.1), le nombre de directions de l’exploration géométrique (Cf. section 4.1.3) et le nombre de facteurs d’échelle (Cf. section 4.1.3). Ces facteurs sont choisis pour n’être que des puissances de 2 afin de réduire l’impact sur l’implémentation. En effet, une multiplication ou une division par ces facteurs reviendra alors à des décalages à gauche ou droite des nombres binaires. Enfin, le nombre de facteurs est également investigué puisque sont autorisés 1 à 5 facteurs différents. Toutes les valeurs sont balayées sur une plage assez large de telle sorte que les augmenter ou diminuer plus n’apporte plus de solutions intéressantes.

## 4.2 Analyses de Pareto des améliorations

Détecteur	Paramètre	Valeurs testées
K-Best	$K$	2, 4, 8, 12, 16, 24, 32, 48, 64, 128, 192, 256
Best-first	Taille des listes	(1, 3, 5), (1, 7, 10), (1, 12, 15)
FA chap. 2	$k$	1, 2, 3, 4, 5
	$\gamma$	$2^{-7}$ , $2^{-6}$ , $2^{-5}$ , $2^{-4}$ , $2^{-3}$ , $2^{-2}$ , $2^{-1}$ , 1
	$F$	10, 20, 30, 40, 60, 90, 130, 150, 175, 200, 250, 275
Géométrique	Sous-const.	Oui, Non
	Max it.	0, 1, 2, 3, 4, 5, 6, 7, 8
	$n_d$	1, 2, 3, 4, 5, 6, 7, 8
	$f$	1 à 5 parmi $2^{-3}$ , $2^{-2}$ , $2^{-1}$ , 1, 2, 4

Tableau 4.4) Liste des paramètres testés dans l'analyse de Pareto

### Analyse de Pareto pour une 16-QAM

La figure 4.7 présente les résultats de l'analyse de Pareto pour une 16-QAM et pour les quatre détecteurs considérés et la figure 4.8 présente les mêmes résultats sans le détecteur à parcours d'arborescence stochastique du chapitre 2. Ces figures permettent de prouver que les modifications ont bien conduit à un détecteur géométrique performant pour les 16-QAM. En effet, bien qu'il ne soit pas efficace au sens de Pareto en comparaison du détecteur à parcours d'arborescence stochastique, il produit des points de fonctionnement proches des optimaux. Ces résultats prouvent une amélioration notable puisque la version historique du détecteur géométrique était absolument incapable de détecter des messages ayant des modulations plus complexes que les QPSK.

La figure 4.8 s'intéresse aux performances du détecteur en comparaison aux références de l'état de l'art précédant ces travaux de thèse. Le détecteur géométrique est efficace au sens de Pareto pour plusieurs points de fonctionnement bien qu'il soit dépassé par le détecteur k-best pour les canaux très bruités et par best-first autour de 20dB de SNR.

### Analyse de Pareto pour une QPSK

La figure 4.9 présente l'analyse de Pareto dans le cas d'une QPSK pour le détecteur géométrique ainsi que pour les deux références de l'état de l'art. Le détecteur à parcours d'arborescence stochastique n'est pas présenté car il n'est pas capable d'atteindre un BER de  $10^{-4}$  pour moins de 20dB de SNR. Cette limite avait déjà été relevée dans le chapitre 2. Le détecteur géométrique montre ici des performances remarquables en étant efficace sur la quasi-totalité de la plage d'utilisation. Le détecteur k-best est le seul détecteur de référence à rester préférable lorsque le canal est assez bruité (8dB) pour justifier de multiplier par plus de 3 la complexité du détecteur. Notre nouveau détecteur permet d'obtenir les mêmes performances de détection pour une complexité 10 fois plus faible que le détecteur best-first à 11dB de SNR et permet de gagner 2dB à complexité constante.

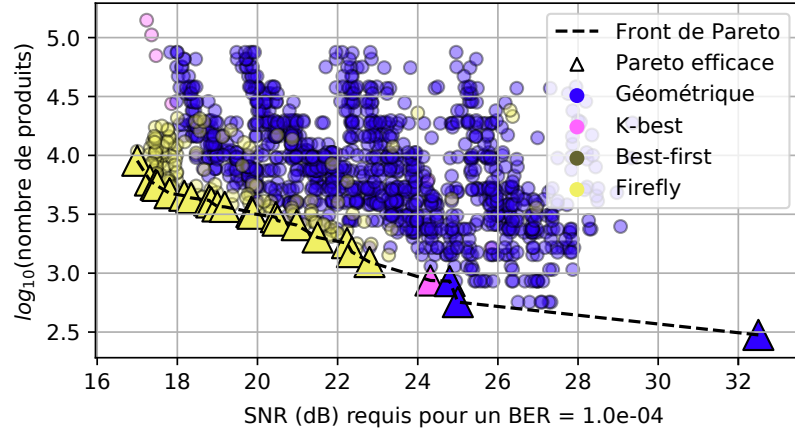


Figure 4.7) Analyse de Pareto des quatre détecteurs pour une 16-QAM.

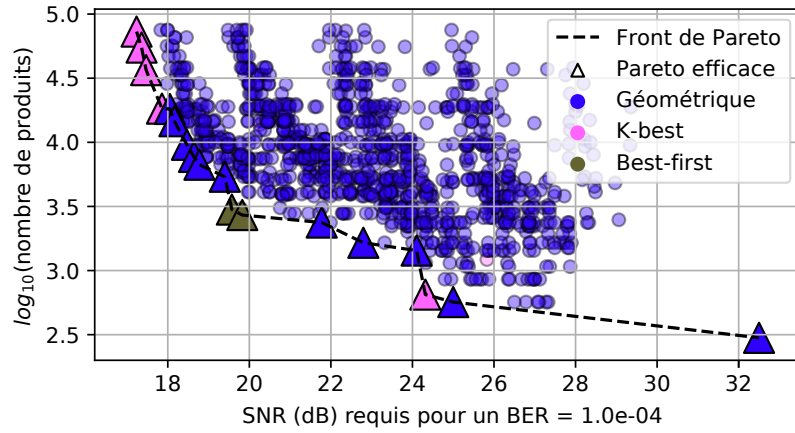


Figure 4.8) Analyse de Pareto sans le détecteur du chapitre 2 pour une 16-QAM.

#### Réduction du nombre de paramètres en QPSK

Cette section propose de réduire le nombre de paramètres de réglage de la même façon que nous l'avions fait pour le détecteur stochastique en section 2.5.2. Cette fois encore, nous cherchons à réduire le nombre de paramètres de réglage tout en conservant les performances et la versatilité du détecteur.

La figure 4.10 présente la même analyse de Pareto mais en ne conservant qu'un seul paramètre de réglage. Pour cette figure, les points de fonctionnement bleus correspondent tous à un détecteur projetant sur les sous-constellations, avec deux facteurs  $f \in \{0.25, 4\}$  et sans la moindre itération de recherche locale. Le chiffre à côté de chaque réglage efficace correspond au nombre  $n_d$  de dimensions utilisées pour la recherche globale. Le cas  $n_d = 0$  revient à construire  $\mathcal{P}$  en projetant uniquement  $\mathbf{x}_0$  sur toutes les sous-constellations.

Ces résultats révèlent tout d'abord que fixer trois des quatre paramètres de réglage n'impacte quasiment pas les qualités de l'algorithme. En effet, la comparaison des figures 4.9 et 4.10 montre que seuls deux points de fonctionnement ne sont pas accessibles. De plus, ces points ne représentaient pas un grand intérêt car ils n'amélioraient que marginalement le SNR pour un surcoût de complexité considérable. Ainsi, le détecteur géométrique peut être efficacement réglé par un seul paramètre en QPSK : le nombre de directions. Ce paramètre unique le rend aussi simple à régler que les k-best tout en étant plus performant. De plus, il s'agit d'un atout considérable pour l'implémentation puisque le nombre de modifications à réaliser en fonction du niveau de bruit du canal est très faible.

Un autre enseignement de cette étude est que la recherche locale n'est plus nécessaire avec les améliorations proposées à la recherche globale. En effet, les LLR peuvent désormais être calculés directement à partir de l'ensemble  $\mathcal{P}$ . Cela signifie que l'exploration avec les projections sur les sous-constellations a capté assez de diversité pour estimer les LLR correctement. De plus, les informations sur la matrice de canal sont suffisamment portées par les vecteurs singuliers  $1/\lambda_i \mathbf{v}_i$  pour ne pas avoir besoin d'une exploitation supplémentaire.

### 4.3 Implémentation matérielle par synthèse haut-niveau

Cette section présente une implémentation matérielle du nouveau détecteur géométrique sur une architecture matérielle. Le FPGA retenu est le Zynq xc72020 de Xilinx.

La section commence par présenter le cas d'utilisation et le paramètre retenu en section 4.3.1 avant de présenter les conséquences de ces choix sur l'implémentation en section 4.3.2. La quantification retenue et l'outil de synthèse haut-niveau (high-level synthesis, HLS) sont par la suite présentés en section 4.3.3 et 4.3.4. Enfin, la section 4.3.5 présente l'architecture de notre détecteur. Enfin, la section 4.3.6 compare les résultats avec des implémentations de références.

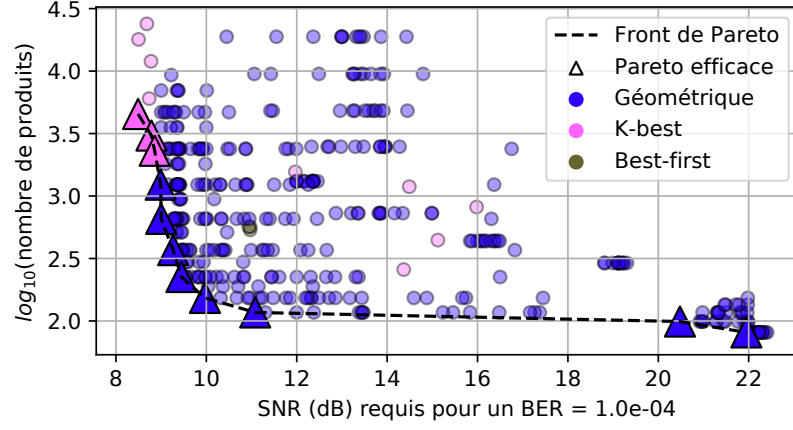


Figure 4.9) Analyse de Pareto sans le détecteur du chapitre 2 pour une QPSK.

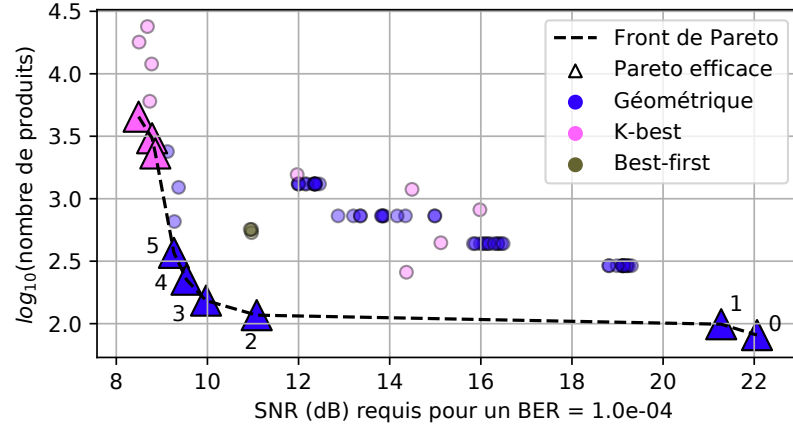


Figure 4.10) Analyse de Pareto pour une QPSK, le détecteur géométrique n'utilise aucune exploitation et  $f \in \{0.25, 4\}$ . Les nombres correspondent au nombre  $n_d$  de directions utilisées pour la recherche globale.



### 4.3.1 Sélection des paramètres et des cas d'utilisation

L'implémentation proposée comporte l'ensemble des opérations niveau vecteur et suppose que les opérations du niveau bloc sont réalisées en amont par une autre partie du FPGA ou par un co-processeur. Cette hypothèse est cohérente tant que les opérations niveau bloc sont négligeables, ce que nous avons supposé depuis le début de ce manuscrit. Dans ce cas de figure, il est primordial d'optimiser les opérations niveau vecteur qui représentent le goulot d'étranglement du détecteur et les opérations niveau bloc qui peuvent être implémentées plus simplement. Cette hypothèse est utilisée par de nombreuses publications de la littérature telles que [4, 82, 83].

L'implémentation est également limitée aux cas d'utilisation pour lesquels les études algorithmiques montrent des atouts en comparaison du reste de l'état de l'art. Ainsi, nous nous concentrons sur les AMC correspondants aux canaux de transmission très bruités décrits dans la section 1.2.1. Nous supposons donc une sortie souple, une QPSK et un canal 4x4 qui peuvent convenir à la fois à la WiFi et au WiMAX.

Enfin, nous fixons les paramètres en accord avec les résultats de la section 4.2.2 puisque cela facilite l'implémentation et le réglage sans impact sur les performances. Ainsi, nous utiliserons deux facteurs d'échelle  $f$  fixés à 0.25 et 4 et nous n'implémenterons pas de recherche locale. Enfin, la projection sur les sous-constellations est mise en place. Le seul paramètre de réglage sera donc  $n_d$  et nous l'autoriserons à varier de 2 à 5 inclus pour couvrir tous les cas de la figure 4.10.

### 4.3.2 Évaluation optimisée de la fonction objectif des points de $\mathcal{P}$

La restriction des cas d'utilisation aux constellations QPSK permet de nouvelles optimisations qui doivent être intégrées pour produire une implémentation efficace. En particulier, la projection sur les sous-constellations est simplifiée car il n'y a alors que quatre sous-constellations possibles listées dans le tableau 4.2. Cette section présente les optimisations permises par cette restriction aux constellations QPSK et la façon dont la fonction objectif est calculée.

Pour illustrer les propos à venir, le tableau 4.5 donne un exemple qui servira de support aux explications. Il présente dans la première case un exemple de liste de candidats avant projection comprenant quatre candidats différents. Ces quatre candidats sont ensuite projetés sur la constellation et sur chacune des sous-constellations pour produire 20 points. Les zones en gras mettent en évidence les changements imposés par les sous-constellations par rapport à la projection sur la constellation totale. La source des optimisations possibles réside dans le fait que projeter sur une sous-constellation revient à forcer la moitié des composantes à une valeur connue et identique pour tous les candidats. Ainsi, connaissant les projetés sur  $\mathcal{Q}$ , l'obtention des autres projections est particulièrement aisée.

La section 4.1.4 a mis en évidence que la fonction objectif pouvait s'exprimer comme la somme de colonnes de  $\mathbf{H}$  signées par la valeur du candidat projeté :

$$\mathbf{H}\mathbf{x} = \sum_{i=1}^{N_t} \mathbf{h}_i x_i \quad (4.7)$$

4 Améliorations, études et implémentation de la détection géométrique

Sous-constellation	Candidats							
Candidats initiaux	0.4	1.3	-0.3	1	0.5	0.5	0.5	-0.4
	-0.1	-0.5	-0.6	-0.6	-0.1	-0.8	-0.5	0.5
	0.8	-0.5	1.2	0.5	8.4	20.2	-6.7	-14
	8.5	8.5	-11	-4	0.3	8.5	-8.5	-20.1
$\mathcal{Q}$	1	1	-1	1	1	1	1	-1
	-1	-1	-1	-1	-1	-1	-1	1
	1	-1	1	1	1	1	-1	-1
	1	1	-1	-1	1	1	-1	-1
$\mathcal{Q}_{1,0}$	-1	-1	-1	-1	1	1	1	-1
	-1	-1	-1	-1	-1	-1	-1	1
	-1	-1	-1	-1	1	1	-1	-1
	-1	-1	-1	-1	1	1	-1	-1
$\mathcal{Q}_{1,1}$	1	1	1	1	1	1	1	-1
	1	1	1	1	-1	-1	-1	1
	1	1	1	1	1	1	-1	-1
	1	1	1	1	1	1	-1	-1
$\mathcal{Q}_{2,0}$	1	1	-1	1	-1	-1	-1	-1
	-1	-1	-1	-1	-1	-1	-1	-1
	1	-1	1	1	-1	-1	-1	-1
	1	1	-1	-1	-1	-1	-1	-1
$\mathcal{Q}_{2,1}$	1	1	-1	1	1	1	1	1
	-1	-1	-1	-1	1	1	1	1
	1	-1	1	1	1	1	1	1
	1	1	-1	-1	1	1	1	1

Tableau 4.5) Exemples de projections sur les sous-constellations d'un QPSK

avec  $\mathbf{h}_i$  la  $i^{\text{ème}}$  colonne de  $\mathbf{H}$  et  $x_i \in \{-1, 1\}$ . Or le tableau 4.5 met en lumière une structure en deux blocs : la première moitié et la seconde moitié des coordonnées de candidats. L'équation (4.7) donne

$$\mathbf{H}\mathbf{x} = \sum_{i=1}^{N_t/2} \mathbf{h}_i x_i + \sum_{i=N_t/2+1}^{N_t} \mathbf{h}_i x_i \quad (4.8)$$

en explicitant les deux blocs. Soient

$$\gamma_{\mathbf{x}} \triangleq \sum_{i=1}^{N_t/2} \mathbf{h}_i x_i \quad \text{et} \quad \zeta_{\mathbf{x}} \triangleq \sum_{i=N_t/2+1}^{N_t} \mathbf{h}_i x_i \quad (4.9)$$

respectivement la somme partielle associée au premier bloc et celle associée au second.

### 4.3 Implémentation matérielle par synthèse haut-niveau

Nous avons alors trivialement,  $\mathbf{H}\mathbf{x} = \gamma_{\mathbf{x}} + \zeta_{\mathbf{x}}$  pour tout  $\mathbf{x}$ . De plus, on définit

$$\gamma_1 \triangleq \sum_{i=1}^{N_t/2} \mathbf{h}_i \quad \text{et} \quad \zeta_1 \triangleq \sum_{i=N_t/2+1}^{N_t} \mathbf{h}_i \quad (4.10)$$

les sommes des  $i^{\text{ème}}$  premières, respectivement dernières, colonnes de  $\mathbf{H}$ .

Avec toutes ces notations, il est possible de construire toutes les sommes possibles à partir des quatre demi-sommes. Le tableau 4.6 liste les combinaisons permettant d'obtenir le produit  $\mathbf{H}\mathbf{x}$  pour chaque projection en sommant un premier et un second terme. De plus,  $\gamma_1$  et  $\zeta_1$  sont indépendants du point à projeter et peuvent donc être calculés au niveau bloc.

Sous-constellation	Premier terme	Second terme
$\mathcal{Q}$	$\gamma_{\mathbf{x}}$	$\zeta_{\mathbf{x}}$
$\mathcal{Q}_{1,0}$	$-\gamma_1$	$\zeta_{\mathbf{x}}$
$\mathcal{Q}_{1,1}$	$\gamma_1$	$\zeta_{\mathbf{x}}$
$\mathcal{Q}_{2,0}$	$\gamma_{\mathbf{x}}$	$-\zeta_1$
$\mathcal{Q}_{2,1}$	$\gamma_{\mathbf{x}}$	$\zeta_1$

Tableau 4.6) Construction des produits  $\mathbf{H}\mathbf{x}$  à partir des demi-sommes.

Cette section a donc permis de simplifier significativement le calcul du produit  $\mathbf{H}\mathbf{x}$  en évitant les calculs redondants. En effet, pour un candidat  $\mathbf{x}$ , seules deux demi-sommes ( $\gamma_{\mathbf{x}}$  et  $\zeta_{\mathbf{x}}$ ) doivent être évaluées pour pouvoir obtenir les  $\mathbf{H}\mathbf{x}$  de toutes ses projections. De plus, nous avons montré que deux demi-sommes ( $\gamma_1$  et  $\zeta_1$ ) peuvent être pré-calculées au niveau bloc.

#### 4.3.3 Quantification des variables

Les implémentations matérielles nécessitent la représentation en virgule fixe des entrées, des sorties et des variables intermédiaires car les opérations sur des nombres flottants sont trop coûteuses. Cependant, il est nécessaire de représenter assez fidèlement les nombres pour éviter une dégradation des performances. Ainsi, la partie entière doit être assez grande pour éviter les débordements (*overflows*) et la partie fractionnaire assez large pour que la précision des décimales soit suffisante. Pour autant, plus les nombres seront représentés par de petits mots binaires, meilleurs sera l'implémentation en termes de débit mais aussi de consommation énergétique. C'est pourquoi, nous avons cherché à réduire au maximum la taille des mots binaires sans perdre en performance.

Le tableau 4.7 présente la quantification retenue pour chaque variable en discriminant les entrées, les sorties et les variables intermédiaires notées *variables*. Nous pouvons noter que tous les nombres sont représentés par moins de 16 bits, soit à titre de comparaison, plus petit qu'un `short int` de C/C++. Afin de s'assurer que la quantification retenue est assez précise, une simulation de Monte-Carlo compare les

performances de la version avec et sans quantification sur la figure 4.11. La version quantifiée perd moins de 0.5dB de SNR sur la version non quantifiée ce qui est suffisant.

Valeur	Type	Taille en bits		
		Entière	Fractionnaire	Total
$\mathbf{H}$	Entrée	2	7	9
$\mathbf{H}^+$	Entrée	4	11	15
$\mathbf{y}$	Entrée	4	3	7
Sommes de (4.5)	Entrée	9	6	15
$\gamma_1$ et $\zeta_1$	Entrée	3	12	15
LLR	Sortie	7	0	7
$\mathbf{x}_0$	Variable	7	5	12
Candidats	Variable	9	6	15
$\gamma_{\mathbf{x}}, \zeta_{\mathbf{x}}$ et $\mathbf{H}\mathbf{x}$	Variable	3	12	15
Fonction objectif	Variable	6	9	15

Tableau 4.7) Quantification des variables.

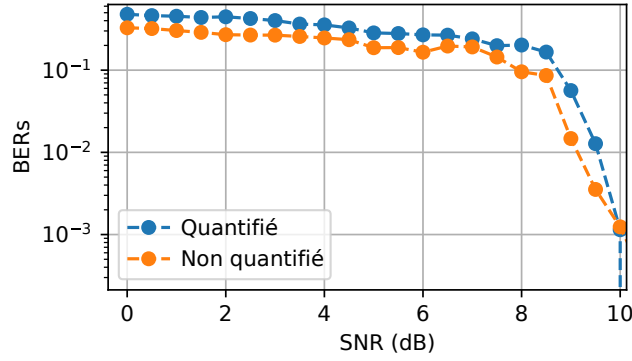


Figure 4.11) Performances avec et sans quantification (QPSK, canal 4x4,  $n_d = 2$ ).

#### 4.3.4 Présentation de la synthèse haut-niveau

La synthèse haut-niveau (hight-level synthesis, HLS) regroupe en ensemble d'outils permettant de convertir un code écrit dans un langage haut-niveau en un langage de description matérielle compatible avec l'implémentation sur FPGA ou ASIC. De nombreux outils de ce type ont été proposés différant les uns des autres par les langages d'entrée et de sortie ainsi que par les optimisations qu'ils proposent [84]. La HLS permet de réduire le temps de développement et de faciliter l'exploration des solutions d'implémentation tout en impactant assez faiblement sur les performances finales.

### 4.3 Implémentation matérielle par synthèse haut-niveau

L'implémentation proposée dans ce chapitre est réalisée grâce à l'outil propriétaire *Vivado HLS* vendu par Xilinx. Les codes d'entrées sont écrits en C++ et des **pragmas** sont ajoutés pour guider la conversion en VHDL ou Verilog. Un code de test en C++ permet de valider que le code C++ correspond au détecteur visé. Enfin, des métriques, telle que le débit, la latence ou la consommation, sont disponibles après la HLS et après le placement-routage par l'outil *Vivado design tool*. L'ensemble des logiciels sont configurés avec les paramètres par défaut.

À titre d'illustration, le code suivant permet d'affecter la valeur `f_obj` à une case du tableau 2D `min` en fonction de la valeur d'une case de `cand`. Le **pragma** demande l'utilisation d'un pipeline sur la boucle `for`. La HLS convertira ce code en une description matérielle incluant les affectations, les conditions ainsi que les contrôles de boucles.

```
for(uint8 coord=0; coord < nb_ant; coord++) {  
#pragma HLS PIPELINE  
    if (cand[coord] == 1) min[coord][1] = f_obj;  
    else min[coord][0] = f_obj;  
}
```

La figure 4.12 donne un extrait du circuit obtenu après HLS où sont omis les multiplexeurs sélectionnant la première dimension de `min` et la gestion de la boucle pour des raisons de lisibilité. Il est important de noter que lorsqu'une condition est présente dans le code d'origine, les circuits correspondants à chaque option sont implémentés et des multiplexeurs permettent de sélectionner le résultat en fonction de cette condition. Dans notre exemple, les deux circuits d'affectation sont présents et la valeur de `cand` commande les multiplexeurs. Ce comportement explique la mauvaise implémentabilité des programmes conditionnels qui demandent d'instancier toutes les options possibles.

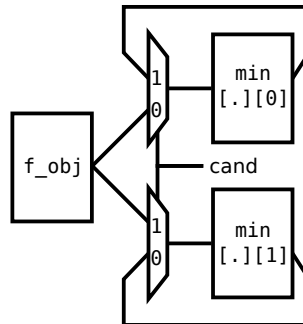


Figure 4.12) Circuit partiel correspondant à l'exemple précédent après HLS

#### 4.3.5 Architecture de l'implémentation

La structure générale de l'implémentation et les parallélisations de chaque module sont décidées de façon itérative selon une méthode classique. Après une première

implémentation naïve, le module limitant est optimisé en allouant plus de ressources jusqu'à ce qu'il ne soit plus le goulot d'étranglement. On réitère le processus jusqu'à atteindre les performances voulues ou qu'il ne reste plus de ressources disponibles. De plus, un fonctionnement en *pipeline* est utilisé dès que possible.

La figure 4.13 présente l'architecture générale retenue. On retrouve les étapes fondamentales présentées précédemment et les entrées correspondantes. Tout d'abord,  $\mathbf{x}_0$  est calculé selon (1.11) à partir de  $\mathbf{H}^+$  et  $\mathbf{y}$ . Les sommes de (4.5) sont ensuite ajoutées pour former les candidats. Ces derniers sont projetés sur  $\mathcal{Q} = \{-1, 1\}$ , ce qui revient à prendre le signe de chaque composante. Les redondances sont ensuite retirées avant de calculer les demi-sommes  $\gamma_{\mathbf{x}}$  et  $\zeta_{\mathbf{x}}$  à partir des colonnes  $\mathbf{h}_i$  de  $\mathbf{H}$  et des points. Enfin, tous les  $\mathbf{y} - \mathbf{H}\mathbf{x}$  sont construits à partir de  $\gamma_{\mathbf{x}}$ ,  $\zeta_{\mathbf{x}}$ ,  $\gamma_1$  et  $\zeta_1$  comme expliqué dans le tableau 4.6. Toutes ces étapes sont implémentées avec des *pipelines* internes mais sans parallélisation car il ne s'agit pas des étapes limitantes.

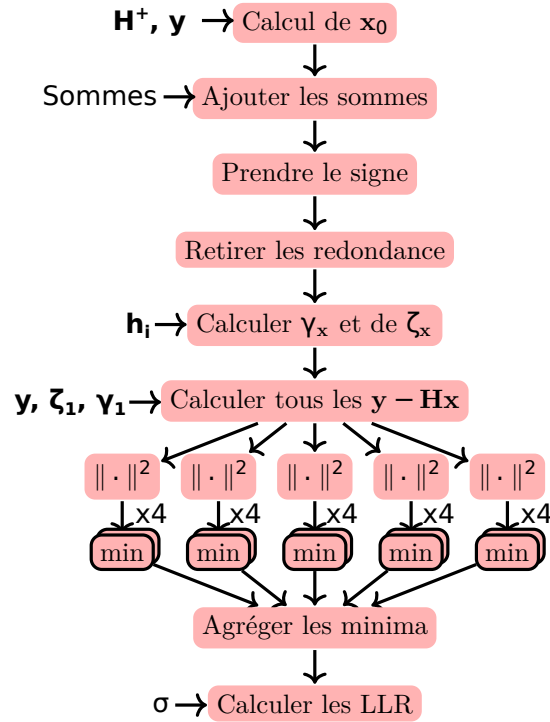


Figure 4.13) Architecture de l'implémentation.

Il ne reste plus qu'à calculer les fonctions objectifs à partir des  $\mathbf{y} - \mathbf{H}\mathbf{x}$  et de trouver les minimums globaux et les contre-hypothèses pour calculer (1.25). La recherche des minima suivis par les calculs de normes sont les deux étapes les plus coûteuses dans cet ordre pour les raisons suivantes.

- Le module `min` classe les candidats entre  $\chi_i^0$  et  $\chi_i^1$  puis recherche les minima

dans chacun des ensembles. Il s'agit de l'opération la plus coûteuse car elle est hautement conditionnelle et nous avons vu dans la section 4.3.4 l'impact des conditions. En effet, la classification requiert une première série de conditions pour placer le candidat dans le bon ensemble. Ensuite, la recherche de minima implique des affectations conditionnelles en fonctions de comparaisons. Ainsi, le module `min` doit implémenter les circuits pour toutes les options possibles, ce qui représente un coût considérable.

- Le calcul des normes requiert  $N_t$  produits par candidat pour passer au carré les coefficients du vecteur avant de les sommer. Or les produits sont réputés être des opérations particulièrement coûteuses à implémenter.

Ces raisons poussent à implémenter cinq instances parallèles de chacun de ces modules. Ainsi, la première branche considère les points projetés sur  $\mathcal{Q}$  tandis que les quatre autres s'occupent des quatre sous-constellations. De plus, le coût encore plus élevé de module `min` impose une parallélisation supplémentaire pour ne pas être limitant. Ainsi, quatre instances du modules sont implémentées en parallèle, chacune traitant 2 des 8 paires  $(\chi_i^0, \chi_i^1)$ , c'est à dire traitant 2 des 8 bits transmis par vecteur  $\mathbf{y}$ .

Les minima de chaque branche sont enfin agrégés et l'approximation max-log (1.25) peut être appliquée pour obtenir les LLR.

#### 4.3.6 Résultats de l'implémentation

L'évaluation des performances d'une implémentation peut passer par plusieurs métriques antagonistes. Dans cette section, nous présentons l'utilisation des ressources, le débit, la consommation énergétique et le comportement du pipeline.

Pour chacune de ces métriques, une comparaison avec des références de la littérature est proposée. Pour autant, cette comparaison n'est pas évidente en l'absence de travaux utilisant la même plateforme dans les mêmes conditions. Ainsi, la majorité des implémentations sont réalisées sur ASIC [4, 59, 62, 82] alors que les études sur FPGA sont beaucoup plus rares [83]. Or, il a été démontré qu'un écart de performance considérable existait entre les deux plateformes matérielles [85, 86]. La taille de la gravure impacte de la même façon les performances des implémentations. Le tableau 4.8 propose des facteurs multiplicatifs approximatifs permettant de passer d'un FPGA gravé en 28nm tel que le nôtre à un ASIC en 65nm qui est utilisé couramment comme référence sur la base de [85, 86]. En plus de la technologie matérielle, le choix de la modulation, de la taille du canal et du type de sorties (dures ou souples) sont d'autres facteurs très impactants susceptibles de brouiller les comparaisons.

##### Utilisation des ressources

La figure 4.14 représente la répartition des ressources entre les différents modules après placement par Vivado. La légende précise la composition des macro-modules relativement aux opérations présentées en figure 4.13. Les données circulent en sens horaire, en commençant par le calcul de  $\mathbf{x}_0$ , puis la création des candidats, les calculs

	FPGA 28nm	ASIC 28nm	ASIC 65nm
Débit ( $\sim f_{clk}$ )	1	3 to 6	1.3 to 2.6
Utilisation des ressources	1	1/30 to 1/13	1/5.6 to 1/2.4
Consommation dynamique	1	1/14 to 1/12	1/6 to 1/5

Tableau 4.8) Facteur multiplicatif des performances en fonction de la technologie.

de  $(\gamma_x, \zeta_x)$  et de la fonction objectif et en finissant par la gestion des minima. Le calcul des LLR n'est pas visible en raison de la très faible surface occupée. En effet, ce calcul est négligeable une fois que les ensembles sont connus comme le précisait le tableau 4.3.

Le macro-module MIN est le plus volumineux ce qui s'explique par la forte parallélisation mise en place au sein de ce dernier. De même, le second module le plus impactant est OBJ FUNC qui contient également des parallélisations supplémentaires.

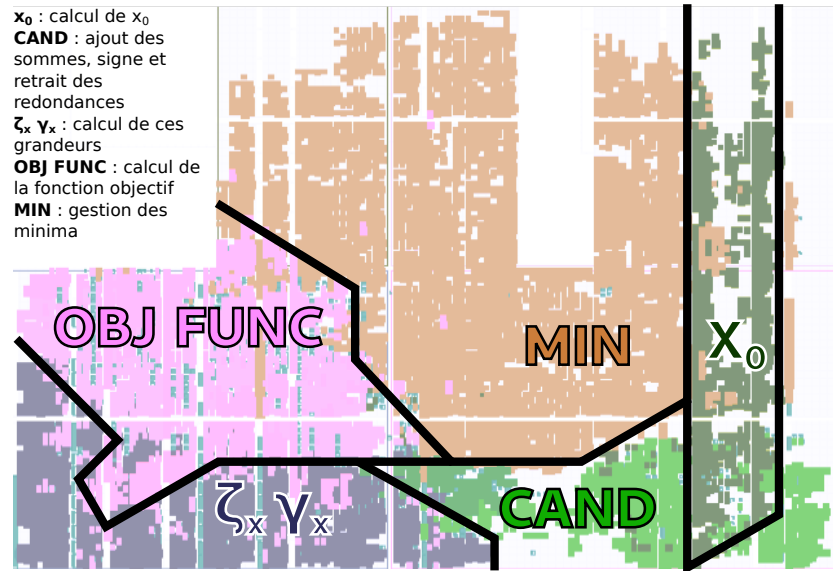


Figure 4.14) Utilisation du FPGA après placement.

Le tableau 4.9 décompte le nombre de ressources utilisées selon le type : table de correspondance (lookup table, LUT), bascule (flip-flop, FF), blocs de mémoire vive (random-access memory, RAM) et processeur de signaux numériques (digital signal processor, DSP). Le nombre absolu est fourni et il est également rapporté à la quantité de ressources matérielles disponibles sur le FPGA utilisé. Enfin, la petite proportion



### 4.3 Implémentation matérielle par synthèse haut-niveau

d'étude réalisée sur FPGA complice beaucoup la comparaison avec la littérature.

Une publication récente rapporte des utilisations de l'ordre de 10 fois inférieure mais sur un canal 2x2 et non 4x4, ainsi que pour des sorties dures plutôt que souples [83]. Or, passer d'un canal 2x2 à 4x4 double approximativement la quantité de ressources nécessaires pour implémenter le détecteur [82]. Ainsi, l'écart d'utilisation n'est plus que d'un facteur 5 avec la référence, écart qui est en partie expliqué par la différence dans le type de sortie. Une autre publication rapporte des consommations similaires aux nôtres mais pour une implémentation détectant les 64-QAM [4].

Type	LUT	FF	BRAM	DSP
Utilisée	16 729	17 451	66	100
% sur xc72020	31	16	47	45

Tableau 4.9) Utilisation des ressources après synthèse haut-niveau.

#### Débit et comportement du *pipeline*

Notre implémentation est cadencée par une horloge à 120MHz ce qui est dans les standards des implémentations sur FPGA [4, 83]. Le tableau 4.10 fournit le débit maximal de chacun des modules à cette fréquence d'horloge. Ces valeurs permettent tout d'abord de déterminer le débit maximal du détecteur qui est égal au minimum des débits des modules : 20Mb/s. Ce débit est inférieur à la littérature. En effet, les implémentations FPGA rapportent des débits de l'ordre de la centaine de Mb/s [4, 83]. De plus, le tableau prouve que les modules de normes et des recherches de minima ont été suffisamment parallélisés puisqu'ils ne sont plus limitants. Le nouveau goulot d'étranglement est le module calculant  $\gamma_{\mathbf{x}}$  et de  $\zeta_{\mathbf{x}}$  mais nous ne disposons pas d'assez de ressources sur le FPGA pour paralléliser ce dernier.

Module	Débit (Mb/s)
Calcul de $\mathbf{x}_0$	37
Ajout des sommes	39
Prendre le signe	100
Retrait des redondances	109
Calcul de $\gamma_{\mathbf{x}}$ et de $\zeta_{\mathbf{x}}$	20
Calcul des $\mathbf{y} - \mathbf{H}\mathbf{x}$	30
$\ \cdot\ ^2$	52
Recherche des minima	37
Agrégation des minima	67
Calcul des LLR	109

Tableau 4.10) Débit maximal de chaque module.

### Consommation énergétique

Vivado estime la consommation de notre détecteur à 677mW dont 118mW de consommation statique et 112mW pour générer l'horloge. De plus, la connaissance du débit permet de calculer une efficacité de l'ordre de 28nJ/b. Grâce au facteur multiplicateur de la table 4.8, on obtient une consommation dynamique d'environ 100mW et une efficacité énergétique d'environ 5nJ/b.

Les références sur ASIC rapportent des consommations dynamiques proches de celles que nous obtenons avec notre implémentation [4, 59, 62, 82]. Pour autant, le débit plus faible que beaucoup de références implique que notre efficacité énergétique est plus élevée d'un facteur proche de 10.

## 4.4 Conclusion sur les améliorations du détecteur géométrique

Dans ce chapitre, nous avons proposé de nombreuses améliorations algorithmiques au détecteur géométrique historique du chapitre précédent. Ces améliorations ont permis de résoudre les trois défauts que nous avons relevés. Tout d'abord, la complexité asymptotique du détecteur croît désormais quadratiquement avec la taille du système de la même façon que les références de l'état de l'art, ce qui favorise l'utilisation du détecteur proposé à des systèmes de tailles élevées. De plus, notre nouvel algorithme peut détecter des transmissions utilisant des constellations 16-QAM, ce que la variante historique était incapable de faire. Enfin, nous avons réduit drastiquement la complexité en QPSK par l'utilisation de pré-calculs au niveau bloc, par la réduction du nombre de candidats et par la suppression de la recherche locale.

Les analyses de Pareto montrent que notre nouveau détecteur est beaucoup plus performant que toutes les références de la littérature pour les signaux modulés en QPSK. Or ce cas de figure apparaît dans plusieurs standards pour transmettre à travers un canal particulièrement bruité. De plus, l'efficacité du détecteur dans ce cas de figure montre une nouvelle fois l'intérêt de l'exploration. En effet, la recherche locale n'est plus nécessaire pour obtenir des performances.

Une implémentation matérielle sur FPGA a enfin été proposée grâce à la HLS et une comparaison avec les références de l'état de l'art a été menée. Le débit et l'efficacité énergétique de notre implémentation sont en dessous des références d'un facteur proche de 5. Ainsi, un travail supplémentaire est nécessaire pour faire tendre les performances de l'implémentation vers les performances théoriques attendues à l'issue de l'étude algorithmique.

## 5 Synthèse des travaux et perspectives

Ce chapitre propose une synthèse des travaux présentés dans ce manuscrit ainsi qu’une conclusion générale. De plus, quelques perspectives de recherches sont proposées sur la base des pistes mises en lumière par nos différentes contributions.

### 5.1 Synthèse des travaux et contributions

Nous avons tout d’abord présenté et analysé l’ensemble du contexte entourant la détection MIMO. La mise en perspective historique a mis en lumière la recherche constante d’améliorations des performances, que ce soit en termes de robustesse, d’efficacité ou de débit de transmission. Cet historique a également montré que l’augmentation des performances s’accompagne toujours d’une complexité accrue du matériel et des techniques de télécommunications. L’analyse des systèmes MIMO et de l’état de l’art a démontré la difficulté du problème de détection qui demande d’approximer le résultat d’un problème NP-dur de façon fiable, rapide et économe sur des plateformes matérielles aux contraintes propres.

La littérature propose déjà de nombreuses stratégies que nous avons classées sur un axe exploration-exploitation. À un opposé de cet axe, on retrouve les détecteurs n’adoptant qu’un comportement d’exploitation. Ces derniers démontrent des résultats remarquables mais peuvent être mis en difficulté lorsque les modèles ou les estimations sont imparfaits ou lorsque les conditions de transmission sont particulièrement dégradées. À l’autre opposé de l’axe se trouvent les détecteurs se concentrant principalement sur l’exploitation. Il s’agit alors de métaheuristiques appliquées au problème de détection en n’exploitant peu ou pas du tout les informations spécifiques aux transmissions MIMO. Cependant, très peu d’études et de détecteurs se placent dans une position intermédiaire de cet axe en mélangeant exploration et exploitation. C’est pourquoi nous avons décidé d’investiguer les opportunités offertes par ce mélange des stratégies, tant au niveau algorithmique qu’au niveau matériel.

Dans un premier temps, nous avons évalué la pertinence du mélange des stratégies en comparant un détecteur mixte avec un détecteur n’effectuant que de l’exploitation. Pour nous assurer que l’ajout d’exploration était bien le facteur influant l’amélioration des performances, nous avons construit un nouveau détecteur à parcours d’arborescence intégrant de l’exploration et étant strictement identique à la littérature pour tout le reste. Pour ce faire, nous avons adapté l’algorithme des lucioles (firefly algorithm, FA) pour produire un nouveau détecteur à parcours d’arborescence stochastique implémentant le compromis exploration-exploitation dans le choix de ses paramètres. Les

analyses de Pareto ont par la suite démontré la supériorité de ce détecteur tant en terme de flexibilité qu'en terme de performances ou de complexité. Ces résultats ont démontré par la même occasion l'intérêt d'investiguer plus largement le mélange de l'exploration et de l'exploitation pour résoudre le problème de détection.

Nous avons ensuite étudié le détecteur géométrique historique qui a la particularité d'implémenter le compromis exploration-exploitation d'une façon très originale. En effet, ce compromis est au centre de la conception du détecteur plutôt que d'être un ajout tardif comme dans le cas précédent. Nous avons longuement évalué ces qualités et ces défauts au travers de calculs de complexité asymptotiques, d'analyse de Pareto ainsi qu'avec un prototype sur SoC. Ainsi, l'ensemble de l'axe algorithmie-implémentation est balayé et nous avons pu mettre en lumière les axes d'améliorations permettant d'atteindre les standards de l'état de l'art : la réduction de la complexité asymptotique, la généralisation du détecteur pour accepter toutes les modulations et l'amélioration des performances en QPSK.

Nous avons ensuite présenté diverses améliorations permettant d'atteindre les objectifs fixés, principalement en travaillant sur la recherche globale et sur la complexité. Toutes ces améliorations ont été ensuite validées par de nouvelles analyses de Pareto qui prouvent que le détecteur géométrique peut détecter les modulations d'ordres supérieurs et qu'il surpasse largement les approches standards sur les transmissions codées utilisant une constellation QPSK. En effet, notre détecteur se montre plus versatile et beaucoup plus efficace que les parcours d'arborescence classique tout en restant aussi simple à régler. Enfin, une implémentation sur FPGA a permis de jauger les qualités de ce détecteur sur architecture matérielle. Les performances mises en lumière sont en deçà de l'état de l'art mais restent assez proches pour suggérer qu'un effort supplémentaire sur l'implémentation permettrait d'atteindre les références de la littérature.

## 5.2 Conclusion

La figure 5.1 positionne nos deux nouveaux détecteurs sur l'axe exploration-exploitation et ajoute les détecteurs de référence de la littérature à titre de comparaison. Ce schéma illustre parfaitement notre volonté d'explorer le régime intermédiaire entre exploration et exploitation.

Nos études précédentes ont montré l'intérêt de se déplacer le long de cet axe en fonction de la qualité du modèle, de l'estimation et du niveau de bruit dans le cadre de standards utilisant des AMC. En effet, lorsque les modèles et les estimations sont extrêmement fiables et que le niveau de bruit est très faible, les standards préfèrent des modulations d'ordre très élevées. Or, les détecteurs à exploitation pure fonctionnent très bien pour ces points de fonctionnement et l'exploration n'est pas nécessaire. De l'autre côté de l'axe, les métaheuristiques agnostiques des modèles sont les plus performantes quand les modélisations et les estimations sont quasiment inutilisables.

Cependant, nous avons proposés deux détecteurs dosant différemment le compromis exploration-exploitation pour les cas intermédiaires. Ainsi, l'ajout d'exploration aux parcours d'arborescence classiques améliore les performances lorsque que les conditions

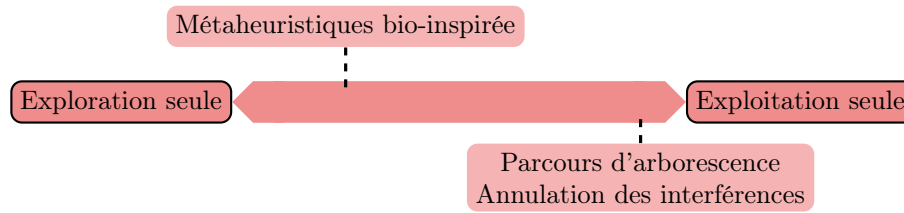


Figure 5.1) Positionnement des références et des nouveaux détecteurs sur l'axe exploration-exploitation. Les références sont sous l'axe, les nouveautés au dessus.

de transmission sont dégradées au point de privilégier les 16-QAM. Cela se traduit par le choix de notre détecteur à parcours d'arborescence stochastique. De plus, lorsque le niveau de bruit est tel que seul une QPSK peut être utilisée, le détecteur géométrique et son exploration plus prononcée fournissent les meilleurs performances tout en exploitant les estimations disponibles.

## 5.3 Perspectives

Cette section propose quelques pistes de recherche possibles issues des contributions et conclusions présentées dans ce manuscrit.

Une première étude complémentaire serait de voir dans quelle mesure la résilience face au bruit se transpose à d'autres perturbations. En effet, nos travaux ont jusqu'à présent évalué la robustesse d'un algorithme à sa capacité à détecter des flux de données plus ou moins bruités. Nous avons ainsi pu prouver que le compromis exploration-exploitation est une bonne façon d'ajuster la stratégie au niveau de bruit. Il serait particulièrement intéressant de voir l'impact du compromis face à des erreurs d'estimation ou de modèle ou encore dans le cas de canaux de transmissions corrélés.

La deuxième piste que nous proposons est l'amélioration de l'implémentation matérielle du nouveau détecteur géométrique. En effet, les analyses algorithmiques suggèrent que ce détecteur devrait être bien plus performant que les références pour les modulations QPSK. Cependant, nous n'avons pas pu transposer ces performances sur plate-forme matérielle. Une implémentation sur ASIC plutôt que sur FPGA est une des étapes nécessaire pour obtenir une comparaison pertinente avec la littérature. De plus, de nombreuses améliorations matérielles devraient permettre d'améliorer le débit de transmission. Citons par exemple une meilleure utilisation des *pipelines* ou une meilleure gestion des lectures-écritures des variables.

Enfin, la recherche d'autres algorithmes mixant exploration et exploitation pourrait mener à des détecteurs plus performants. En effet, nous n'avons proposé ici que deux stratégies dont une adaptée du parcours d'arborescence classique. Il est tout à fait possible que l'ajout d'exploration sous d'autre forme et/ou à d'autres détecteurs puisse conduire à des performances supérieures. De même, une toute nouvelle stratégie construite autour de ce compromis pourrait utiliser de façon plus efficace ce compromis.

## *5 Synthèse des travaux et perspectives*

En particulier, il serait très appréciable de trouver un détecteur unique capable de parcourir l'axe exploration-exploitation plutôt que de devoir changer radicalement de stratégie en fonction du point de fonctionnement (qualité du canal et de l'estimation, présence de corrélation, validité des modèles, etc).

# Liste des acronymes

- 2G** deuxième génération
- 3G** troisième génération
- 4G** quatrième génération
- 5G** cinquième génération
- ACO** optimisation par colonies de fourmis (ant colony optimizations)
- AMC** codage et modulation adaptatifs (adaptive coding and modulation)
- ASIC** circuit intégré spécifique à une application (application-specific integrated circuit)
- BER** taux d'erreurs binaires (bit error ratio)
- BFS** parcours en largeur (breadth-first search)
- BnB** séparation et évaluation (branch and bound)
- BP** propagation des convictions (belief propagation)
- BPSK** modulation par changement de phase binaire (binary phase-shift keying)
- CA** automate cellulaire (cellular automaton)
- CORDIC** calcul numérique par rotation de coordonnées (coordinate rotation digital computer)
- DFS** parcours en profondeur (depth-first search)
- DSP** processeur de signaux numériques (digital signal processor)
- FA** algorithme des lucioles (firefly algorithm)
- FF** bascule (flip-flop)
- FPGA** réseau de portes programmables (field-programmable gate array)
- HLS** synthèse haut-niveau (high-level synthesis)
- IC** annulation des interférences (interferences cancellation)
- LDPC** contrôle de parité de faible densité (low-density parity-check)

## 5 Synthèse des travaux et perspectives

- LLR** ratio des log-vraisemblances (log-likelihood ratio)
- LOS** ligne de vue direct (line-of-sight)
- LTE** long-term evolution
- LUT** table de correspondance (lookup table)
- MBF** parcours par métrique modifié (modified best-first)
- MBF-FD** parcours par métrique modifié avec descente rapide (modified best-first with fast descent)
- MCS** modulation and coding scheme
- MFS** parcours selon la métrique (metric-first search)
- MIMO** à entrées et sorties multiples (multiple-input multiple-output)
- ML** maximum de vraisemblance (maximum likelihood)
- MMSE** à erreur quadratique moyenne minimale (minimum mean square error)
- MU-MIMO** MIMO à utilisateurs multiples (multi-user MIMO)
- NLOS** sans ligne de vue (non-line-of-sight)
- NOMA** multiplexage non orthogonal (non-orthogonal multiple access)
- OSIC** annulation successive ordonnée des interférences (ordered successive interferences cancellation)
- PED** distance euclidienne partielle (partial Euclidian distance)
- PIC** annulation des interférences en parallèle (parallel interferences cancellation)
- QAM** modulation d'amplitude en quadrature (quadrature amplitude modulation)
- QPSK** modulation par changement de phase en quadrature (quadrature phase-shift keying)
- RAM** mémoire vive (random-access memory)
- ROM** mémoire morte (read-only memory)
- SD** détection sphérique (sphere decoding)
- SE** Schnorr-Euchner
- SIC** annulation successive des interférences (successive interferences cancellation)
- SINR** rapport signal à interférences plus bruit (signal-to-interferences-plus-noise ratio)
- SIR** rapport signal à interférences (signal-to-interferences ratio)



**SISO** à entrée et sortie unique (single-input single-output)

**SNR** rapport signal à bruit (signal-to-noise ratio)

**SoC** système sur puce (system on chip)

**SPCA** automate cellulaire auto-programmable (self-programmable cellular automaton)

**SVD** décomposition en valeurs singulières (singular values decomposition)

**VHDL** Very high speed integrated circuit Hardware Description Language

**ZF** forçage à zéro (zero-forcing)



## Liste des références

- [1] BASTIEN TROTOBAS, AMOR NAFKHA, and YVES LOUËT. “A Review to Massive MIMO Detection Algorithms: Theory and Implementation”. en. In: *Advanced Radio Frequency Antennas for Modern Communication and Medical Systems*. Intechopen, July 2020 (cit. on pp. 1, 17).
- [2] ARIJIT DATTA and VIMAL BHATIA. “A near maximum likelihood performance modified firefly algorithm for large MIMO detection”. en. In: *Swarm and Evolutionary Computation* 44 (Feb. 2019), pp. 828–839. ISSN: 22106502. DOI: [10.1016/j.swevo.2018.09.004](https://doi.org/10.1016/j.swevo.2018.09.004) (cit. on pp. 2, 33, 36, 45, 47, 62, 90).
- [3] ZHAN GUO and PETER NILSSON. “Algorithm and implementation of the K-best sphere decoding for MIMO detection”. en. In: *IEEE Journal on Selected Areas in Communications* 24.3 (Mar. 2006), pp. 491–503. ISSN: 0733-8716. DOI: [10.1109/JSAC.2005.862402](https://doi.org/10.1109/JSAC.2005.862402) (cit. on pp. 2, 29, 36, 42, 44, 52, 54, 59, 73, 74, 90).
- [4] GUANGHUI HE, XIAOYU ZHANG, and ZHUOJUN LIANG. “Algorithm and Architecture of an Efficient MIMO Detector With Cross-Level Parallel Tree-Search”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28.2 (Feb. 2020). Conference Name: IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 467–479. ISSN: 1557-9999. DOI: [10.1109/TVLSI.2019.2947495](https://doi.org/10.1109/TVLSI.2019.2947495) (cit. on pp. 2, 31, 36, 42, 44, 52, 53, 56, 58, 59, 90, 95, 101, 103, 104).
- [5] BASTIEN TROTOBAS, YOUNESS AKOURIM, AMOR NAFKHA, and YVES LOUËT. “Adding Exploration to Tree-Based MIMO Detectors Using Insights from Bio-Inspired Firefly Algorithm”. In: *IEEE Vehicular Technology Conference*. Helsinki, Finland, Apr. 2021 (cit. on pp. 2, 62).
- [6] AMOR NAFKHA. “A geometrical approach detector for solving the combinatorial optimisation problem: Application to wireless communication systems.” Theses. Université de Bretagne Sud, Mar. 2006 (cit. on pp. 2, 34, 36, 63, 67, 68).
- [7] A. NAFKHA, E. BOUTILLON, and C. ROLAND. “Quasi-maximum-likelihood detector based on geometrical diversification greedy intensification”. In: *IEEE Transactions on Communications* 57.4 (Apr. 2009), pp. 926–929. ISSN: 0090-6778. DOI: [10.1109/TCOMM.2009.04.060603](https://doi.org/10.1109/TCOMM.2009.04.060603) (cit. on pp. 2, 34, 36, 63, 67).
- [8] BASTIEN TROTOBAS, YOUNESS AKOURIM, AMOR NAFKHA, YVES LOUËT, and JACQUES WEISS. “Evaluation of the Complexity, Performance and Implementability of Geometrical MIMO Detectors: the Example of the Exploration and Exploitation List Detector”. en. In: *International Journal On Advances in Telecommunications* 13.1&2 (2019), pp. 1–9 (cit. on pp. 2, 68, 71).

## Liste des références

- [9] B. TROTOBAS, A. LLAVE, A. NAFKHA, and Y. LOUËT. “When Should We Use Geometrical-Based MIMO Detection Instead of Tree-Based Techniques? A Pareto Analysis”. In: *IEEE Access* 8 (2020), pp. 191163–191173. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2020.3032016](https://doi.org/10.1109/ACCESS.2020.3032016) (cit. on pp. 2, 79).
- [10] CLAUDE SHANNON and WARREN WEAVER. *The Mathematical Theory of Communication*. en. Urbana: University of Illinois Press, 1949. ISBN: 0-252-72546-8 (cit. on pp. 5, 7).
- [11] MOHAMMAD D. AL-AMRI, MOHAMED EL-GOMATI, and M. SUHAIL ZUBAIRY, eds. *Optics in Our Time*. en. Cham: Springer International Publishing, 2016. ISBN: 978-3-319-31902-5 978-3-319-31903-2. DOI: [10.1007/978-3-319-31903-2](https://doi.org/10.1007/978-3-319-31903-2) (cit. on p. 6).
- [12] STEFANO SELLERI. “Claude Chappe and the first telecommunication network (without electricity)”. In: *URSI Radio Science Bulletin* 360 (2017), pp. 96–101 (cit. on p. 6).
- [13] I. GLESK. “From Morse code to Terabit communications”. In: *IEEE Conference on the History of Communications*. Nov. 2010, pp. 1–6. DOI: [10.1109/HISCON.2010.5735303](https://doi.org/10.1109/HISCON.2010.5735303) (cit. on p. 6).
- [14] W.F. BRINKMAN, D.E. HAGGAN, and W.W. TROUTMAN. “A history of the invention of the transistor and where it will lead us”. en. In: *IEEE Journal of Solid-State Circuits* 32.12 (Dec. 1997), pp. 1858–1865. ISSN: 00189200. DOI: [10.1109/4.643644](https://doi.org/10.1109/4.643644) (cit. on p. 7).
- [15] RICHARD W HAMMING. “Error detecting and error correcting codes”. en. In: *The Bell system technical journal* 29.2 (1950), pp. 147–160 (cit. on p. 7).
- [16] HARRY NYQUIST. “Certain topics in telegraph transmission theory”. In: *Transactions of the American Institute of Electrical Engineers* 47.2 (1928), pp. 617–644 (cit. on p. 7).
- [17] CLAUDE E SHANNON. “Communication in the Presence of Noise”. en. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21 (cit. on p. 7).
- [18] THOMAS KAILATH and AROGYASWAMI J. PAULRAJ. “Increasing capacity in wireless broadcast systems using distributed transmission/directional reception (DTDR)”. US5345599A. Sept. 1994 (cit. on p. 7).
- [19] III RICHARD H. ROY and BJORN OTTERSTEN. “Spatial division multiple access wireless communication systems”. US5515378 (A). May 1996 (cit. on p. 7).
- [20] GERARD J. FOSCHINI. “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas”. en. In: *Bell Labs Technical Journal* 1.2 (1996), pp. 41–59. ISSN: 1538-7305. DOI: <https://doi.org/10.1002/bltj.2015> (cit. on p. 7).

- [21] G.G. RALEIGH and J.M. CIOFFI. “Spatio-temporal coding for wireless communication”. en. In: *IEEE Transactions on Communications* 46.3 (Mar. 1998), pp. 357–366. ISSN: 00906778. DOI: [10.1109/26.662641](https://doi.org/10.1109/26.662641) (cit. on p. 7).
- [22] G.D. GOLDEN, C.J. FOSCHINI, R.A. VALENZUELA, and P.W. WOLNIANSKY. “Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture”. en. In: *Electronics Letters* 35.1 (1999), p. 14. ISSN: 00135194. DOI: [10.1049/e1:19990058](https://doi.org/10.1049/e1:19990058) (cit. on p. 7).
- [23] S.M. ALAMOUTI. “A simple transmit diversity technique for wireless communications”. In: *IEEE Journal on Selected Areas in Communications* 16.8 (Oct. 1998), pp. 1451–1458. ISSN: 07338716. DOI: [10.1109/49.730453](https://doi.org/10.1109/49.730453) (cit. on p. 8).
- [24] V. TAROKH, H. JAFARKHANI, and A.R. CALDERBANK. “Space-time block codes from orthogonal designs”. en. In: *IEEE Transactions on Information Theory* 45.5 (July 1999), pp. 1456–1467. ISSN: 00189448. DOI: [10.1109/18.771146](https://doi.org/10.1109/18.771146) (cit. on p. 8).
- [25] F. R. FARROKHI, G. J. FOSCHINI, A. LOZANO, and R. A. VALENZUELA. “Link-optimal space-time processing with multiple transmit and receive antennas”. In: *IEEE Communications Letters* 5.3 (Mar. 2001), pp. 85–87. ISSN: 1089-7798. DOI: [10.1109/4234.913148](https://doi.org/10.1109/4234.913148) (cit. on pp. 8, 12).
- [26] R.W. HEATH and A.J. PAULRAJ. “Switching between diversity and multiplexing in MIMO systems”. In: *IEEE Transactions on Communications* 53.6 (June 2005), pp. 962–968. ISSN: 1558-0857. DOI: [10.1109/TCOMM.2005.849774](https://doi.org/10.1109/TCOMM.2005.849774) (cit. on p. 8).
- [27] THOMAS PAUL and TOKUNBO OGUNFUNMI. “Evolution, insights and challenges of the PHY layer for the emerging IEEE 802.11n amendment”. en. In: *IEEE Communications Surveys & Tutorials* 11.4 (2009), pp. 131–150. ISSN: 1553-877X. DOI: [10.1109/SURV.2009.090408](https://doi.org/10.1109/SURV.2009.090408) (cit. on pp. 8, 10).
- [28] IEEE STD 802.1. “IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput”. In: (Oct. 2009), pp. 1–565. ISSN: 978-0-7381-6046-7. DOI: [10.1109/IEEESTD.2009.5307322](https://doi.org/10.1109/IEEESTD.2009.5307322) (cit. on pp. 8, 10).
- [29] IEEE STD 802.16E. *IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*. Tech. rep. 2005 (cit. on pp. 11, 16, 87, 90).
- [30] S. WU, C. WANG, E. M. AGGOUNE, and M. M. ALWAKEEL. “A novel Kronecker-based stochastic model for massive MIMO channels”. In: *2015 IEEE/CIC International Conference on Communications in China (ICCC)*. Nov. 2015, pp. 1–6. DOI: [10.1109/ICCCChina.2015.7448642](https://doi.org/10.1109/ICCCChina.2015.7448642) (cit. on pp. 11, 13).

## Liste des références

- [31] EMIL BJÖRNSON, ERIK G. LARSSON, and THOMAS L. MARZETTA. “Massive MIMO: Ten Myths and One Critical Question”. en. In: *IEEE Communications Magazine* 54.2 (Feb. 2016). arXiv: 1503.06854, pp. 114–123. ISSN: 0163-6804. DOI: [10.1109/MCOM.2016.7402270](https://doi.org/10.1109/MCOM.2016.7402270) (cit. on pp. 12, 85).
- [32] S. O. RICE. “Mathematical analysis of random noise”. In: *The Bell System Technical Journal* 23.3 (July 1944), pp. 282–332. ISSN: 0005-8580. DOI: [10.1002/j.1538-7305.1944.tb00874.x](https://doi.org/10.1002/j.1538-7305.1944.tb00874.x) (cit. on p. 12).
- [33] NAFISEH SHARIATI and MATS BENGTTSSON. “How Far from Kronecker can a MIMO Channel be? Does it Matter?” en. In: *17th European Wireless 2011-Sustainable Wireless Technologies*. VDE, 2011, pp. 1–7 (cit. on p. 13).
- [34] S.L. LOYKA. “Channel capacity of MIMO architecture using the exponential correlation matrix”. en. In: *IEEE Communications Letters* 5.9 (Sept. 2001), pp. 369–371. ISSN: 1089-7798. DOI: [10.1109/4234.951380](https://doi.org/10.1109/4234.951380) (cit. on p. 13).
- [35] A. M. SAYEED. “Modeling and capacity of realistic spatial MIMO channels”. In: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 4. May 2001, 2489–2492 vol.4. DOI: [10.1109/ICASSP.2001.940506](https://doi.org/10.1109/ICASSP.2001.940506) (cit. on p. 13).
- [36] K. YU, M. BENGTTSSON, B. OTTERSTEN, D. MCNAMARA, P. KARLSSON, and M. BEACH. “Modeling of Wide-Band MIMO Radio Channels Based on NLoS Indoor Measurements”. en. In: *IEEE Transactions on Vehicular Technology* 53.3 (May 2004), pp. 655–665. ISSN: 0018-9545. DOI: [10.1109/TVT.2004.827164](https://doi.org/10.1109/TVT.2004.827164) (cit. on p. 13).
- [37] M.O. DAMEN, A. ABDI, and M. KAVEH. “On the effect of correlated fading on several space-time coding and detection schemes”. en. In: *IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No.01CH37211)*. Vol. 1. Atlantic City, NJ, USA: IEEE, 2001, pp. 13–16. ISBN: 978-0-7803-7005-0. DOI: [10.1109/VTC.2001.956544](https://doi.org/10.1109/VTC.2001.956544) (cit. on p. 13).
- [38] ÖZGECAN ÖZDOĞAN, EMIL BJÖRNSON, and ERIK G. LARSSON. “Uplink Spectral Efficiency of Massive MIMO with Spatially Correlated Rician Fading”. en. In: *arXiv:1805.07968 [cs, math]* (May 2018). arXiv: 1805.07968 (cit. on p. 13).
- [39] HIGH THROUGHPUT TASK GROUP. *TGn Channel Models*. en. Tech. rep. IEEE Std. 802.11-03/940r4. IEEE, May 2004 (cit. on p. 13).
- [40] XIANG GAO, OVE EDFORS, FREDRIK RUSEK, and FREDRIK TUFVESSON. “Massive MIMO performance evaluation based on measured propagation data”. en. In: *IEEE Transactions on Wireless Communications* 14.7 (July 2015). arXiv: 1403.3376, pp. 3899–3911. ISSN: 1536-1276. DOI: [10.1109/TWC.2015.2414413](https://doi.org/10.1109/TWC.2015.2414413) (cit. on p. 14).

- [41] TODD K. MOON. *Error Correction Coding: Mathematical Methods and Algorithms*. en. 2nd Edition. John Wiley & Sons, 2020. ISBN: 1-119-56749-1 (cit. on pp. [14](#), [72](#)).
- [42] D. MICCIANCIO. “The hardness of the closest vector problem with preprocessing”. en. In: *IEEE Transactions on Information Theory* 47.3 (Mar. 2001), pp. 1212–1215. ISSN: 00189448. DOI: [10.1109/18.915688](#) (cit. on p. [19](#)).
- [43] J. JALDEN and B. OTTERSTEN. “On the complexity of sphere decoding in digital communications”. en. In: *IEEE Transactions on Signal Processing* 53.4 (Apr. 2005), pp. 1474–1484. ISSN: 1053-587X. DOI: [10.1109/TSP.2005.843746](#) (cit. on pp. [19](#), [27](#), [36](#)).
- [44] STEVEN M. KAY. *Fundamentals of statistical signal processing*. Englewood Cliffs, N.J.: Prentice-Hall, 1993. ISBN: 978-0-13-345711-7 (cit. on p. [20](#)).
- [45] P. W. WOLNIANSKY, G. J. FOSCHINI, G. D. GOLDEN, and R. A. VALENZUELA. “V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel”. In: *URSI International Symposium on Signals, Systems, and Electronics*. Oct. 1998, pp. 295–300. DOI: [10.1109/ISSSE.1998.738086](#) (cit. on pp. [21](#), [36](#)).
- [46] A. RIADI, M. BOULOUIRD, and M. M. HASSANI. “ZF/MMSE and OSIC Detectors for UpLink OFDM Massive MIMO systems”. In: *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. Apr. 2019, pp. 767–772. DOI: [10.1109/JEEIT.2019.8717380](#) (cit. on p. [21](#)).
- [47] W.H. CHIN, A.G. CONSTANTINIDES, and D.B. WARD. “Parallel multistage detection for multiple antenna wireless systems”. en. In: *Electronics Letters* 38.12 (2002), p. 597. ISSN: 00135194. DOI: [10.1049/el:20020419](#) (cit. on pp. [22](#), [23](#), [36](#)).
- [48] B.M. HOCHWALD and S. TEN BRINK. “Achieving near-capacity on a multiple-antenna channel”. en. In: *IEEE Transactions on Communications* 51.3 (Mar. 2003), pp. 389–399. ISSN: 0090-6778. DOI: [10.1109/TCOMM.2003.809789](#) (cit. on pp. [23](#), [28](#)).
- [49] CHRISTOPH STUDER, SCHEKEB FATEH, and DOMINIK SEETHALER. “ASIC Implementation of Soft-Input Soft-Output MIMO Detection Using MMSE Parallel Interference Cancellation”. In: *IEEE Journal of Solid-State Circuits* 46.7 (July 2011), pp. 1754–1765. ISSN: 0018-9200, 1558-173X. DOI: [10.1109/JSSC.2011.2144470](#) (cit. on pp. [23](#), [36](#)).
- [50] M. TUCHLER, A.C. SINGER, and R. KOETTER. “Minimum mean squared error equalization using a priori information”. In: *IEEE Transactions on Signal Processing* 50.3 (Mar. 2002), pp. 673–683. ISSN: 1941-0476. DOI: [10.1109/78.984761](#) (cit. on pp. [23](#), [36](#)).

## Liste des références

- [51] W. KOCH and A. BAIER. “Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference”. In: *IEEE Global Telecommunications Conference and Exhibition*. Dec. 1990, pp. 1679–1684. DOI: [10.1109/GLOCOM.1990.116774](#) (cit. on pp. 23, 28).
- [52] P. ROBERTSON, E. VILLEBRUN, and P. HOEHER. “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain”. In: *IEEE International Conference on Communications*. Vol. 2. June 1995, pp. 1009–1013. DOI: [10.1109/ICC.1995.524253](#) (cit. on pp. 23, 28).
- [53] ALESSANDRO TOMASONI, MARCO FERRARI, DEVIS GATTI, FABIO OSNATO, and SANDRO BELLINI. “A Low Complexity Turbo MMSE Receiver for W-LAN MIMO Systems”. In: *IEEE International Conference on Communications*. Vol. 9. ISSN: 1938-1883. June 2006, pp. 4119–4124. DOI: [10.1109/ICC.2006.255726](#) (cit. on pp. 23, 36).
- [54] LINGLONG DAI, SHENG CHEN, and LAJOS HANZO. “A Survey of Non-Orthogonal Multiple Access for 5G”. en. In: *IEEE communications surveys & tutorials* 20.3 (2018), pp. 2294–2323 (cit. on p. 24).
- [55] U. FINCKE and M. POHST. “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis”. en. In: *Mathematics of Computation* 44.170 (1985), pp. 463–471. ISSN: 0025-5718, 1088-6842. DOI: [10.1090/S0025-5718-1985-0777278-8](#) (cit. on pp. 26, 36).
- [56] C P SCHNORR and M EUCHNER. “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems.” en. In: *Mathematical programming* 66.1-3 (1994), pp. 181–199 (cit. on p. 28).
- [57] E. AGRELL, T. ERIKSSON, A. VARDY, and K. ZEGER. “Closest point search in lattices”. en. In: *IEEE Transactions on Information Theory* 48.8 (Aug. 2002), pp. 2201–2214. ISSN: 0018-9448. DOI: [10.1109/TIT.2002.800499](#) (cit. on p. 28).
- [58] C. STUDER and H. BOLCSKEI. “Soft-Input Soft-Output Single Tree-Search Sphere Decoding”. In: *IEEE Transactions on Information Theory* 56.10 (Oct. 2010), pp. 4827–4842. ISSN: 0018-9448. DOI: [10.1109/TIT.2010.2059730](#) (cit. on pp. 28, 36).
- [59] PIERCE I-JEN CHUANG, MANOJ SACHDEV, and VINCENT C. GAUDET. “VLSI implementation of high-throughput, low-energy, configurable MIMO detector”. en. In: *2015 33rd IEEE International Conference on Computer Design (ICCD)*. New York City, NY, USA: IEEE, Oct. 2015, pp. 335–342. ISBN: 978-1-4673-7166-7. DOI: [10.1109/ICCD.2015.7357162](#) (cit. on pp. 28, 36, 101, 104).
- [60] KITTY K Y WONG and PETER J MCLANE. “A Low-Complexity Iterative MIMO Detection Scheme Using the Soft-Output M-Algorithm”. en. In: *IST Mobile Summit* (June 2005), p. 5 (cit. on pp. 29, 36).



- [61] SEUNG-JIN CHOI, SEONG-JOON SHIM, YOUNG-HWAN YOU, JAESANG CHA, and HYOUNG-KYU SONG. “Novel MIMO Detection With Improved Complexity for Near-ML Detection in MIMO-OFDM Systems”. en. In: *IEEE Access* 7 (2019), pp. 60389–60398. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2019.2914707](https://doi.org/10.1109/ACCESS.2019.2914707) (cit. on pp. 29, 36).
- [62] Z. YAN, G. HE, Y. REN, W. HE, J. JIANG, and Z. MAO. “Design and Implementation of Flexible Dual-Mode Soft-Output MIMO Detector With Channel Preprocessing”. In: *IEEE Transactions on Circuits and Systems* 62.11 (Nov. 2015), pp. 2706–2717. ISSN: 1549-8328. DOI: [10.1109/TCSI.2015.2479055](https://doi.org/10.1109/TCSI.2015.2479055) (cit. on pp. 29, 36, 38, 101, 104).
- [63] MUHAMMAD S. KHAIRY, CHUNG-AN SHEN, AHMED M. ELTAWIL, and FADI J. KURDAHI. “Algorithms and Architectures of Energy-Efficient Error-Resilient MIMO Detectors for Memory-Dominated Wireless Communication Systems”. en. In: *IEEE Transactions on Circuits and Systems* 61.7 (July 2014), pp. 2159–2171. ISSN: 1549-8328, 1558-0806. DOI: [10.1109/TCSI.2014.2298273](https://doi.org/10.1109/TCSI.2014.2298273) (cit. on pp. 29, 36).
- [64] EDSGER W DIJKSTRA. “A Note on Two Problems in Connexion with Graphs”. en. In: *Numerische mathematik* 1.1 (1959), pp. 269–271. DOI: [10.1007/BF01386390](https://doi.org/10.1007/BF01386390) (cit. on p. 30).
- [65] DONALD E. KNUTH. *Art of Computer Programming, The Volume 1: Fundamental Algorithms*. 3rd. Vol. 1. Boston, United States: Addison-Wesley Professional, July 1997. ISBN: 978-0-201-89683-1 (cit. on pp. 30, 42).
- [66] CHUN-HAO LIAO, TO-PING WANG, and TZI-DAR CHIUH. “A 74.8 mW Soft-Output Detector IC for 8 x 8 Spatial-Multiplexing MIMO Communications”. In: *IEEE Journal of Solid-State Circuits* 45.2 (Feb. 2010), pp. 411–421. ISSN: 1558-173X. DOI: [10.1109/JSSC.2009.2037292](https://doi.org/10.1109/JSSC.2009.2037292) (cit. on pp. 31, 36).
- [67] NEEV SAMUEL, TZVI DISKIN, and AMI WIESEL. “Deep MIMO detection”. In: *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. ISSN: 1948-3252. July 2017, pp. 1–5. DOI: [10.1109/SPAWC.2017.8227772](https://doi.org/10.1109/SPAWC.2017.8227772) (cit. on pp. 32, 36).
- [68] MATHIEU GOUTAY, FAYÇAL AIT AOUDIA, JAKOB HOYDIS, and JEAN-MARIE GORCE. “Machine Learning for MU-MIMO Receive Processing in OFDM Systems”. en. In: *arXiv:2012.08177 [cs, eess, math]* (Dec. 2020). arXiv: 2012.08177 (cit. on pp. 32, 36).
- [69] MEHRDAD KHANI, MOHAMMAD ALIZADEH, JAKOB HOYDIS, and PHIL FLEMING. “Adaptive Neural Signal Detection for Massive MIMO”. en. In: *IEEE Transactions on Wireless Communications* 19.8 (2020). arXiv: 1906.04610, pp. 5635–5648 (cit. on pp. 32, 36).

## Liste des références

- [70] JOSÉ CARLOS MARINELLO and TAUFİK ABRÃO. “Lattice Reduction Aided Detector for MIMO Communication Via Ant Colony Optimisation”. en. In: *Wireless Personal Communications* 77.1 (July 2014), pp. 63–85. ISSN: 0929-6212, 1572-834X. DOI: [10.1007/s11277-013-1495-z](https://doi.org/10.1007/s11277-013-1495-z) (cit. on pp. 32, 36, 38).
- [71] XIN-SHE YANG. “Firefly Algorithm, Stochastic Test Functions and Design Optimisation”. en. In: *International journal of bio-inspired computation* 2.2 (Mar. 2010). arXiv: 1003.1409, pp. 78–84 (cit. on pp. 33, 36, 38, 45).
- [72] RICHARD S. SUTTON and ANDREW G. BARTO. *Reinforcement Learning: An introduction*. en. MIT Press, 2018 (cit. on p. 35).
- [73] WEI-CHENG SUN, WEI-HSUAN WU, CHIA-HSIANG YANG, and YEONG-LUH UENG. “An Iterative Detection and Decoding Receiver for LDPC-Coded MIMO Systems”. In: *IEEE Transactions on Circuits and Systems* 62.10 (Oct. 2015), pp. 2512–2522. ISSN: 1549-8328, 1558-0806. DOI: [10.1109/TCSI.2015.2468998](https://doi.org/10.1109/TCSI.2015.2468998) (cit. on p. 38).
- [74] PULI ANIL KUMAR. “FPGA Implementation of the Trigonometric Functions Using the CORDIC Algorithm”. en. In: *International Conference on Advanced Computing & Communication Systems (ICACCS)*. Coimbatore, India: IEEE, Mar. 2019, pp. 894–900. ISBN: 978-1-5386-9531-9 978-1-5386-9533-3. DOI: [10.1109/ICACCS.2019.8728315](https://doi.org/10.1109/ICACCS.2019.8728315) (cit. on p. 48).
- [75] LUCIAN PETRICA. “FPGA optimized cellular automaton random number generator”. en. In: *Journal of Parallel and Distributed Computing* 111 (Jan. 2018), pp. 251–259. ISSN: 0743-7315. DOI: [10.1016/j.jpdc.2017.05.022](https://doi.org/10.1016/j.jpdc.2017.05.022) (cit. on p. 49).
- [76] GENE H. GOLUB and WILLIAM KAHAN. “Calculating the singular values and pseudo-inverse of a matrix”. In: *SIAM Journal on Numerical Analysis*. Society for Industrial and Applied Mathematics 2.2 (1965), pp. 205–224 (cit. on p. 65).
- [77] GENE H. GOLUB and C REINSCH. “Singular value decomposition and least squares solutions”. en. In: *Numerische Mathematik*. Springer 14.5 (1970), pp. 403–420 (cit. on p. 65).
- [78] GENE H. GOLUB and CHARLES F. VAN LOAN. *Matrix computations*. en. 4. ed. Johns Hopkins studies in mathematical sciences. OCLC: 824733531. Baltimore, Md: Johns Hopkins Univ. Press, 2013. ISBN: 978-1-4214-0794-4 978-1-4214-0859-0 (cit. on pp. 65, 67).
- [79] EMIL BJÖRNSON, JAKOB HOYDIS, and LUCA SANGUINETTI. *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*. en. 2017 (cit. on p. 66).
- [80] M. KRSTIC, E. GRASS, and X. FAN. “Asynchronous and GALS Design -Overview and Perspectives”. In: *New Generation of CAS (NGCAS)*. Sept. 2017, pp. 85–88. DOI: [10.1109/NGCAS.2017.42](https://doi.org/10.1109/NGCAS.2017.42) (cit. on p. 77).

- [81] DAVID E. MULLER. *Theory of asynchronous circuits*. eng. Urbana, Illinois : University of Illinois, Graduate College, Digital Computer Laboratory, Dec. 1955 (cit. on p. 78).
- [82] ESSI SUIKKANEN and MARKKU JUNTTI. “Performance and ASIC Designs of the K-best LSD and LMMSE Detectors for LTE Downlink”. en. In: *Journal of Signal Processing Systems* 91.11-12 (Dec. 2019), pp. 1291–1304. ISSN: 1939-8018, 1939-8115. DOI: [10.1007/s11265-019-1441-8](https://doi.org/10.1007/s11265-019-1441-8) (cit. on pp. 95, 101, 103, 104).
- [83] A SHIRLY EDWARD and S MALARVIZHI. “Architectural implementation of modified K-best algorithm for detection in MIMO systems”. en. In: *Microprocessors and Microsystems*. Elsevier 74 (2020), p. 103010 (cit. on pp. 95, 101, 103).
- [84] RAZVAN NANE et al. “A Survey and Evaluation of FPGA High-Level Synthesis Tools”. en. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.10 (Oct. 2016), pp. 1591–1604. ISSN: 0278-0070, 1937-4151. DOI: [10.1109/TCAD.2015.2513673](https://doi.org/10.1109/TCAD.2015.2513673) (cit. on p. 98).
- [85] I. KUON and J. ROSE. “Measuring the Gap Between FPGAs and ASICs”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 26.2 (Feb. 2007), pp. 203–215. ISSN: 0278-0070. DOI: [10.1109/TCAD.2006.884574](https://doi.org/10.1109/TCAD.2006.884574) (cit. on p. 101).
- [86] ANDREW BOUTROS, SADEGH YAZDANSHENAS, and VAUGHN BETZ. “You Cannot Improve What You Do not Measure: FPGA vs. ASIC Efficiency Gaps for Convolutional Neural Network Inference”. en. In: *ACM Transactions on Reconfigurable Technology and Systems* 11.3 (Dec. 2018), pp. 1–23. ISSN: 19367406. DOI: [10.1145/3242898](https://doi.org/10.1145/3242898) (cit. on p. 101).



---

**Titre :** Compromis exploration-exploitation dans la détection MIMO : algorithmie et implémentation

**Mots clés :** Détection MIMO, Implémentation FPGA, Compromis exploration-exploitation

**Résumé :** Un système MIMO (multiple-inputs, multiple-outputs) transmet plusieurs flux de données dans le même slot temps-fréquence entre émetteur(s) et récepteur disposant de plusieurs antennes. Cette démultiplication des flux de données améliore le débit, la robustesse et l'efficacité des transmissions au prix d'une réception beaucoup plus ardue. En effet, il faut alors séparer les flux de données en plus de compenser les défauts du canal de transmission (bruits et évanouissements). C'est le rôle des algorithmes de détection.

La grande majorité des détecteurs modernes exploitent les modèles ainsi que les estimations de l'état du canal de transmission pour retrouver les signaux émis.

Cependant, ces approches sont en difficultés lorsque le canal est trop bruité, mal estimé ou trop différent des modèles attendus. D'autres détecteurs bio-inspirés explorent plus largement les solutions possibles pour éviter ces défauts au risque de réaliser plus de calcul et donc de dégrader les performances.

Ce manuscrit analyse le compromis entre exploration et exploitation et développe des algorithmes permettant de mélanger ces deux stratégies. Pour ce faire, deux détecteurs sont développés : l'un mixant les exploitations traditionnelles avec l'exploration bio-inspirée de l'algorithme des lucioles, l'autre à partir d'une approche nouvelle. Les études de ce manuscrit s'intéressent tant à l'algorithmie des détecteurs qu'à leurs implémentations matérielles.

---

**Title :** Exploration-exploitation trade-off in MIMO detection: algorithms and implementation

**Keywords :** MIMO detection, FPGA implementation, Exploration-exploitation trade-off

**Abstract :** A MIMO (multiple-inputs, multiple-outputs) system transmits, with multiple antennas, several data streams in the same time-frequency. The increasing number of data streams enhances the throughput, the reliability and the efficiency of such transmissions at the expense of a much more challenging reception. Indeed, it is required to separate the data streams and to counteract the inherent degradation of the transmission channel (noise and fading). This is the role of detection algorithms.

Most modern detectors exploit models and estimates of the state of the transmission channel to retrieve the transmitted signals. However, these approaches struggle when the channel is too noisy, badly estimated or deviates

too much from the models. Other bio-inspired detectors explore more comprehensively the possible solutions to overcome these issues at the cost of more computation and consequently degraded performance.

This manuscript discusses the trade-off between exploration and exploitation and designs algorithms to mix these two strategies. For this purpose, two detectors are developed: one mixing the traditional exploitations with the bio-inspired exploration of the firefly algorithm, the other one based on a novel approach. These studies cover both the algorithmic aspects of the detectors and the hardware implementations.