Habilitation à Diriger les Recherches de l'Université de Bretagne Occidentale

Vers de Nouvelles Applications du Codage de Canal

Elsa Dupraz
elsa.dupraz@imt-atlantique.fr

Soutenance le 18 octobre 2023 devant le jury composé de :

Laurent Clavier	Professeur, IMT Nord Europe	Rapporteur
Pascal Giard	Professeur, École de Technologie Supérieure	Rapporteur
Jean-Marie Gorce	Professeur, INSA Lyon	Rapporteur
Emmanuel Boutillon	Professeur, Université de Bretagne Sud	Examinateur
Catherine Douillard	Professeur, IMT Atlantique	Examinatrice
Roland Gautier	Professeur, Université de Bretagne Occidentale	Examinateur
Giuseppe Valenzise	Chargé de Recherche, Université Paris-Saclay	Examinateur

Sommaire

Chapitre 1

CV

1.1 Parcours

1.1.1 Études

2010 - 2013 : Doctorat de l'Université Paris-Saclay, intitulé "Codage de sources avec information adjacente incertaine au décodeur", soutenu le 3 décembre 2013, devant le jury composé de :

- Enrico Magli, Associate Professor à Politechnico de Torino (président du jury)
- Charly Poulliat, Professeur à INP - ENSEIHT (rapporteur)
- Vladimir Stankovic, Senior Lecturer à Université de Strathclyde (rapporteur)
- Pierre Duhamel, Directeur de recherche CNRS (examinateur)
- Claudio Weidmann, Maître de conférences Université de Cergy-Pontoise (examinateur)
- Michel Kieffer, Professeur Université Paris-Sud (directeur de thèse)
- Aline Roumy, Chargée de Recherche INRIA (co-directrice de thèse)

2009-2010 : Master SAR (Systèmes Avancés de Radiocommunications) en télécommunications, co-accrédité par par l'ENS Cachan, Supélec, et l'université Paris-Saclay, et obtention du diplôme de l'ENS Cachan.

2007 - 2009 : Département EEA (Electronique, Electrotechnique, Automatique) **de l'ENS Cachan**, License et Master 1 IST (Information, Systèmes, Technologie) de l'Université Paris-Saclay.

1.1.2 Parcours professionnel

Depuis 2015 : Enseignante-Chercheuse au département SC (Signal et Communications) puis MEE (Mathematical and Electrical Engineering) de l'IMT Atlantique (ex

Telecom Bretagne).

Activités de recherche en théorie de l'information, codage canal, codage de sources, traitement de signal, machine learning. Activités d'enseignement en probabilités, statistiques, machine learning, méthodes numériques, optimisation, codage de sources, codage de canal.

2013 - 2015 : Post-doctorat à l'Université d'Arizona aux Etats-Unis et à l'ENSEA/laboratoire ETIS en France. Activités de recherche sur la conception de codes, encodeurs, et décodeurs LDPC implémentés sur des architectures matérielles non-fiables.

2010 - 2013 : Thèse de doctorat au L2S (Laboratoire des Signaux et Systèmes), laboratoire de l'Université Paris-Sud, Supélec, et du CNRS. **Mission d'enseignement** à l'Ecole Polytechnique.

Activités de recherche en théorie de l'information et codes LDPC, avec appplication au codage de sources avec information adjacente au décodeur. Activités d'enseignement en traitement du signal et traitement du son, encadrement de projets.

Avril - Juillet 2010 : Stage de recherche de Master 2 au L2S (Laboratoire des Signaux et Systèmes), intitulé "Codage de sources avec information adjacente au décodeur pour un modèle de source de Markov caché" et encadré par Michel Kieffer et Francesca Bassi.

Juin - Juillet 2009 : Stage de recherche de Master 1 au département Traitement de Signal du LTCI (Laboratoire Traitement et Communication de l'Information) à Télécom ParisTech, intitulé "Réalisation d'un algorithme de fingerprint audio robuste au changement de vitesse de lecture" et encadré par Gaël Richard.

1.2 Activités de recherche

Les outils que j'utilise dans mes recherches proviennent des domaines de la théorie de l'information, du codage de canal, avec en particulier l'utilisation de codes LDPC, du traitement de signal, et du Machine Learning. Dans la plupart de mes travaux, je cherche à mettre en oeuvre une démarche qui consiste à partir de la théorie, avec l'étude d'une version modélisée du problème, pour ensuite concevoir des schémas de codage source/canal d'abord génériques, puis dédiés à une application. La partie application se fait le plus souvent en collaboration avec des chercheurs experts du domaine, qui pourront fournir leur expertise et leurs données sur le sujet.

Dans cette démarche, la théorie de l'information sert à obtenir les performances at-

teignables par un système de communications, et fournit souvent des indications pour la conception du système de codage. Par exemple, l'étude théorique pourra dire s'il est optimal ou de non de réaliser le codage sources et le codage canal de manière séparés. Ensuite, les outils des domaines du codage de canal, du traitement du signal, et du Machine Learning, me servent à concevoir des solutions pratiques de codage. En particulier, en codage de sources, étant donné que l'on traite des données, on a besoin de méthodes avancées en signal et en ML.

Pendant mes premières années à l'IMT Atlantique, j'ai appliqué cette démarches à deux axes de recherche principaux : (i) l'utilisation de codes LDPC pour le codage de sources, (ii) la conception de décodeurs LDPC à très faible consommation d'énergie. J'ai obtenu des financements pour plusieurs projets sur chacun de ces axes. Le projet Cominlabs InterCom ainsi qu'un projet projet PHC Pavle Savic m'ont permis de travailler sur l'axe (i). Le projet ANR JCJC EF-FECtive m'a permis de développer une activité importante sur l'axe (ii), et de développer des collaborations internationales qui ont donné lieu au projet AI-EF, en collaboration avec l'Université de l'Illinois à Urbana-Champain, et au projet REFined, avec Polytechnique Montréal.

A la fin de ces projets, j'ai commencé à travailler sur de nouveaux sujets qui me permettent d'utiliser et de continuer à développer mes compétences dans les domaines cités précédemment. En particulier, je continue à explorer les liens entre codage et Machine Learning, à travers plusieurs problèmes : utilisation du machine learning pour la conception de codes LDPC (projet AI4CODE), conception de systèmes de codage source/canal pour des applications en Machine Learning (projets CoLearn et IoTAD-CEO). J'ai aussi commencé à travailler sur un problème plus spécifique au domaine du codage canal : la conception de codes correcteurs d'erreurs pour le stockage de données dans l'ADN.

1.2.1 Projets de recherche

Dans cette partie, je décris les projets de recherche nationaux et internationaux auxquels j'ai participé depuis mon arrivée à l'IMT Atlantique en 2015.

Projets nationaux en cours

CoLearn, septembre 2021 - décembre 2024 (coordinatrice) : Projet de recherche du Labex Cominlabs, en partenariat avec l'INSA Rennes et l'INRIA Rennes. Jiahui Wei (doctorant) travaille sur ce projet dans le cadre d'un co-encadrement entre

l'IMT Atlantique et l'INSA Rennes.

Ce projet porte sur la conception de codes source/canal pour des applications de type apprentissage automatique. En particulier, nous supposerons que le serveur central, qui reçoit l'ensemble des informations collectées par des capteurs, souhaite soit reconstruire les données, soit appliquer une tâche d'apprentissage sur ces données. Nous souhaitons traiter deux questions fondamentales, que sont l'étude du compromis entre les deux critères d'intérêt, et l'optimialité de la séparation du codage source et du codage canal dans cette configuration. Nous traiterons ces questions avec des outils de théorie de l'information, et travaillerons à la conception de schémas de codage pratiques pour ce problème.

AI4CODE, novembre 2021 - octobre 2025 : Projet de recherche de type ANR PRC, en partenariat avec le CEA Leti, l'INP Bordeaux, l'INP-ENSEEIHT, le laboratoire ETIS, et l'UBS. A l'IMT Atlantique, Alireza Tasdighi (post-doctorant) a travaillé sur ce projet, et Ahmad Ismail (doctorant) va travailler sur ce projet.

Le projet porte sur l'utilisation d'outils d'intelligence artificielle pour la conception de codes correcteurs d'erreurs et des décodeurs associés. Plus spécifiquement, nous allons travailler sur l'utilisation d'outils de type re-inforcement learning pour la conception de codes LDPC, et sur des approches de type Deep Learning pour l'amélioration des performances de décodeurs LDPC. Dans les deux cas, nous viserons une application à la communication de trames courtes de données.

DnarXiv, octobre 2020 - décembre 2022 : Projet de recherche du Labex Cominlabs, en partenariat avec l'INRIA Rennes, l'UBS, le LATIM, et l'IGDR. Dans le cadre de ce projet, je participe à l'encadrement de la thèse de Belaïd Hamoum, doctorant à l'UBS.

Le projet porte sur le stockage de données dans des molécules d'ADN. A partir de données expérimentales collectées dans le cadre du projet, nous avons développé un modèle statistique de canal de stockage de données dans l'ADN, prenant en compte la mémoire et la dépendance à la séquence d'entrée dans les erreurs d'insertions, de déletions, de substitutions, introduites lors des opérations de synthèse (écriture) et de séquençage (lecture) de l'ADN. Nous travaillons maintenant à la conception de codes correcteurs d'erreurs (codes LDPC et codes convolutifs) permettant d'éliminer les erreurs introduites par ces opérations.

Projets nationaux terminés

EF-FECtive, janvier 2018 - décembre 2021 (coordinatrice) : Projet de Recherche de type ANR JCJC. À l'IMT Atlantique, Mohamed Yaoumi (doctorant) et Jo-

nathan Kern (doctorant) ont travaillé sur le projet. Ce projet a donné lieu à une collaboration active avec Polytechnique Montréal (Canada) et avec l'Université de l'Illinois à Urbana-Champain (États-Unis).

Le projet portait sur la conception de codes et décodeurs LDPC à très faible consommation d'énergie. Pour cela, nous avons considéré des décodeurs LDPC implémentés sur du matériel sous-alimenté, ce qui permet de réduire la consommation d'énergie du circuit, mais introduit des erreurs dans les unités de calcul et dans les mémoires. Nous avons proposé des modèles statistiques réalistes pour prédire la consommation d'énergie du circuit. Puis nous avons intégré ces modèles d'énergie dans les outils de conception de codes LDPC, ce qui nous a permis d'optimiser les paramètres du code et du décodeur de manière à minimiser leur consommation d'énergie. Enfin, nous avons développé une architecture matérielle flexible permettant de tester et valider expérimentalement l'approche proposée dans le projet.

InterCom, novembre 2016 - octobre 2020 (responsable pour l'IMT Atlantique) : Projet de recherche du Labex Cominlabs, en partenariat avec l'INRIA Rennes, l'IFSTTAR, et le L2S. À l'IMT Atlantique, Fangping Ye (doctorant), Zeina Mheich (post-doctorante), Mai Quyen Pham (post-doctorante) ont travaillé sur le projet.

Le projet portait sur le codage de sources pour les communications interactives et l'accès massif et aléatoire à des grandes bases de données. Nous avons tout d'abord utilisé des outils de la théorie de l'information pour prédire la performance optimale atteignable par ces systèmes de codage. Puis nous avons proposé des schémas pratiques de codage, utilisant notamment des codes LDPC. Nous avons ensuite adapté les schémas de codage proposés à une application particulière : le codage d'images à 360°. Enfin, nous avons proposé une représentation en graphes du problème d'accès aléatoire à des ensembles de sources, et développé des méthodes d'optimisation de ces graphes pour minimiser les débits de stockage et de transmission de sous-ensembles de ces sources.

COLA, janvier 2016 - juillet 2017 : Contrat de recherche avec Huawei, en collaboration avec Raphaël Le Bidan et Fréderic Guilloud (IMT Atlantique). Zeina Mheich (post-doctorante à l'IMT Atlantique) a travaillé sur le projet.

Le contrat portait sur la conception de codes LDPC non-binaires pour la communication de trames courtes de données. Nous avons travaillé sur la construction de codes LDPC non-binaires compatibles en rendement, et sur la complexité de décodage des codes LDPC non-binaires.

Projets internationaux en cours

IoTAD-CEO, avril 2021 - décembre 2024 (responsable pour l'IMT Atlantique) : Chaire Internationale financée par Cominlabs et l'IMT Atlantique, portée par Tadashi Matsumoto, chercheur à JAIST, au Japon. A l'IMT Atlantique, Ismaila Salihou Adamou (doctorant) travaille dans le cadre de ce projet.

Ce projet porte sur la conception de systèmes de communication multi-utilisateurs pour la prise de décision. Dans un contexte où un grand nombre de capteurs doivent transmettre leurs données à un centre de fusion, on considère que l'objectif du centre de fusion n'est pas de reconstruire les données, mais de prendre une décision à partir des données reçues. Nous nous intéressons à une méthode de prise de décision très simple : le test d'hypothèses. Pour ce problème, l'objectif est de concevoir le système de codage source/canal pour la prise de décisions, dans des configurations éventuellement complexes de communications : canaux à évanouissement, canaux de type MAC (multiple access channel), présence de relais, etc. Dans le cadre de ce projet, Tadashi Matsumoto a effectué une première visite à l'IMT Atlantique, et reviendra pour plusieurs séjours longs (quelques mois par an) pendant la durée du projet. Amin Zribi, chercheur à IsetCom en Tunisie, est impliqué dans la chaire, et va également effectuer des séjours à l'IMT Atlantique. Tadashi Matsumoto et Amin Zribi sont tous les deux chercheurs associés à l'IMT Atlantique.

REFined, avril 2020 - décembre 2022 (coordinatrice) : projet du programme Samuel de Champlain, en collaboration avec Polytechnique Montréal, Canada.

Le projet porte sur l'étude du compromis entre la consommation d'énergie et la fiabilité des méthodes de Machine Learning travaillant sur des architectures matérielles non-fiables. L'objectif du projet est de déterminer des modèles statistiques réalistes pour relier la consommation d'énergie du circuit à la proportion de fautes introduites dans les algorithmes. Il s'agira ensuite d'exploiter ces modèles pour concevoir des codes correcteurs d'erreurs permettant de corriger les fautes introduites par le circuit. Dans ce cadre, nous nous sommes pour le moment concentrés sur le calcul en mémoire à partir de matrices de memristors, et sur l'implémentation de réseaux de neurones utilisant cette technologie. Jonathan Kern effectue sa thèse en co-tutelle avec Polytechnique Montréal sur ce sujet. Les visites entre partenaires ont pour le moment été limitées à cause du COVID.

Projets internationaux terminés

AI-EF, août 2018 - décembre 2021 (coordinatrice) : Projet du programme Thomas Jefferson "Make our planet great again", financé par la FACE foundation, en collaboration avec l'Université de l'Illinois à Urbana-Champain, aux États-Unis.

Le projet portait sur la conception d'algorithmes de Machine Learning à faible consommation d'énergie, en considérant l'implémentation de ces algorithmes sur des architectures matérielles non-fiables. Dans un premier temps, nous avons travaillé sur des algorithmes récursifs, de types estimation binaire et filtrage de Kalman. Pour ces algorithmes, nous avons étudié la propagation des erreurs introduites par l'architecture au cours des itérations successives de l'algorithme, et proposé des mécanismes de correction. Dans un second temps, nous nous sommes intéréssés au calcul en mémoire à partir de structures de type matrices de memristors. Nous avons étudié l'effet de l'imprécision des valeurs de memristors dans les couches linéaires des réseaux de neurones, et proposé des méthodes d'optimisation de la puissance de ces architecture, sous contrainte de performance. Lav Varshney a effectué une visite à l'IMT Atlantique, et Jonathan Kern, doctorant à l'IMT Atlantique, a travaillé avec lui dans le cadre de ce projet. L'ensemble de la collaboration a été fructueuse mais les visites ont été plus limitées que ce que nous aurions voulu, à cause du COVID.

SEED, janvier 2018 - décembre 2019 (coordinatrice) : Projet de type PHC Pavle Savic, financé par Campus-France, en collaboration avec MISANU en Serbie.

Le projet portait sur le codage de sources pour des réseaux de capteurs à faible consommation d'énergie, avec des outils de théorie de l'information pour l'analyse théorique de performances, et l'utilisation de codes LDPC pour les constructions pratiques de schémas de codages. Dans le cadre de ce projet, Fangping Ye (doctorant), Mohamed Yaoumi (doctorant), et moi-même sommes venus en visite à Belgrade, en Serbie, tandis que Velimir Ilic et Miodrag Mihaljevic ont effectué des séjours à l'IMT Atlantique.

1.2.2 Encadrement

Je recense maintenant les étudiants que j'ai eu la chance d'encadrer pendant leur post-doctorats, doctorats, ou stages.

Post-doctorants

— *Ahcen Aliouat*, juillet 2023 - juin 2024, codage source/canal pour l'apprentissage automatique, co-encadré par François-Xavier Socheleau

— *Alireza Tasdighi*, juin 2020 - mai 2022, codage source/canal pour des applications en Machine Learning, co-encadré par Raphaël Le Bidan

— *Khaled Alhaj Ali*, avril 2020 - décembre 2020, calcul en mémoire pour les réseaux de neurones binaires, co-encadré par Amer Baghdadi et Mathieu Léonardon

— *Mai Quyen Pham*, avril 2018 - octobre 2019, optimisation de graphes pour l'accès aléatoire à des données, co-encadré par Aline Roumy et Thomas Maugey

— *Zeina Mheich*, fevrier 2016 - août 2017, codes LDPC non-binaires, co-encadré par Raphaël Le Bidan et Fréderic Guilloud

Doctorants

— *Aref Ezzeddine*, novembre 2022 - octobre 2025, codage canal pour le stockage de données dans l'ADN, co-encadré par Emmanuel Boutillon

— *Ahmad Ismail*, novembre 2022 - octobre 2025, Deep Learning pour la conception de codes LDPC, co-encadré par Raphaël Le Bidan

— *Ismaila Salihou Adamou*, décembre 2021 - novembre 2024, systèmes de communications multi-utilisateurs pour la prise de décisions, co-encadré par Samir Saoudi et Tadashi Matsumoto

— *Jiahui Wei*, octobre 2021 - septembre 2024, théorie de l'information et codage pour le Machine Learning, co-encadré par Philippe Mary

— *Belaïd Hamoum*, octobre 2019 - décembre 2022, codes correcteurs d'erreurs pour le stockage de données dans l'ADN, thèse soutenue le 13 décembre 2023, co-encadré par Laura Conde-Canencia et Emmanuel Boutillon

— *Jonathan Kern*, octobre 2019 - mai 2023, algorithmes de machine learning à faible consommation d'énergie, thèse soutenue le 5 mai 2023, co-encadré par François Leduc-Primeau et Abdeldjalil Aïssa El Bey

— *Mohamed Yaoumi*, novembre 2017 - décembre 2020, décodeurs LDPC à faible consommation d'énergie, thèse soutenue le 14 décembre 2020, co-encadré par Frederic Guilloud

— *Fangping Ye*, novembre 2016 - décembre 2019, codes LDPC pour le codage de sources, thèse soutenue le 2 décembre 2019, co-encadré par Karine Amis

Stagiaires

— *Tianfeng Lyu*, mai - novembre 2022, étudiant à l'IMT Atlantique, réseaux de neurones implémentés dans des unités de calcul en mémoire

— *Marc-André Lavoie*, juin - août 2020, étudiant à Polytechnique Montréal, décodeurs LDPC implémentés dans des unités de calcul en mémoire, co-encadré par François Leduc-Primeau

— *Elodie Derringer*, juin - juillet 2018, étudiante à l'IMT Atlantique, algorithmes de clustering distribués, co-encadré par Dominique Pastor

— *Sarah El Beji*, juin - juillet 2018, étudiante à l'IMT Atlantique, clustering sur données compressées

— *Salma El Ghourbal*, juin - juillet 2018, étudiante à l'IMT Atlantique, algorithmes d'estimation récursive binaire implémentés sur matériel non-fiable

— *Fangping Ye*, avril - septembre 2016, étudiant à Paris-Sud, codes LDPC pour le codage de sources

— *Guillaume Muret*, juin - juillet 2016, étudiant à l'IMT Atlantique, clustering sur données compressées

— *Ji Wei*, fevrier - août 2013, étudiante à l'université Paris-Sud, protocoles de communications sur hurlement de loups, co-encadré par Pierre Gerold et François Meriaux

— *Zheng Chen*, juin - août 2012, étudiante à l'université Paris-Sud, modèles de corrélation pour le codage vidéo, co-encadré par Michel Kieffer

1.3 Services à la communauté

1.3.1 IEEE Information Theory Society

Depuis janvier 2022, je suis membre du "Digital Presence Online Committee" de la société IEEE Information Theory (ITSoc). Ce comité est responsable de la diffusion en ligne des informations de la société, par le biais du site web, des mailing lists, et des réseaux sociaux.

Je fais aussi partie du comité "Teaching Ressources for Information Theory" d'ITSoc, dont l'objectif est de créer un site web qui contiendra des ressources (slides de cours, polycopiés de cours, feuilles d'exercices, etc.) pour l'enseignement de la théorie de l'information.

1.3.2 Organisation d'évènements scientifiques

Je fais, ou j'ai fait partie des comités d'organisation pour les conférences internationales suivantes :

- *International Symposium on Topics in Coding* (ISTC), Brest, septembre 2023, en tant que conference management chair
- *Information Theory Workshop* (ITW), Kanazawa, Japon, octobre 2021, en tant que Publicity chair
- *International Symposium on Topics in Coding* (ISTC), Montréal, septembre 2021, en tant que conference management chair
- *International Symposium on Turbo Codes and Iterative Information Processing* (ISTC), Brest, septembre 2016, en tant que conference management chair

J'ai aussi participé à l'organisation des journées d'études et sessions spéciales suivantes :

- *Journée GdR ISIS* : "Stockage de données numériques dans l'ADN synthétique", Paris, Juillet 2023
- *Session spéciale* : "DNA data storage", conférence DSP, Grèce, Juin 2023
- *Session spéciale* : "Design of Energy-Efficient Error-Correction Codes", conférence ISTC, Montréal, Septembre 2021
- *Journée GdR ISIS* : "Energy-efficient LDPC decoders", Paris, Juin 2016

1.3.3 Jurys et comités

J'ai participé aux jurys de thèses suivants :

- Soutenance de thèse de Maëlic Louart, 7 juillet 2023, sur le sujet "Conception d'un récepteur AIS détectant les falsifications de message", Ecole Navale, Brest (examinatrice)
- Soutenance de thèse de Muhammad Umar Farooq, 25 novembre 2022, sur le sujet : "Spatial coupling, turbo-like codes, and their connection to LDPC codes", Lund

University, Suède (examinatrice)

— Soutenance de thèse de Khaled Taleb, 19 mai 2022, sur le sujet "Physical layer security : secure communications", ISAE-Supaéro, Toulouse (examinatrice)

— Soutenance de thèse de Franklin Cochachin, le 2 mai 2019, sur le sujet "Noise-against-Noise Decoders : Low Precision Iterative Decoders", UBS, Lorient (examinatrice)

— Soutenance de thèse de Jérome Fellus, le 3 octobre 2017, sur le sujet "Algorithmes décentralisés et asynchrones pour l'apprentissage statistique large échelle et application à l'indexation multimédia", ENSEA, Cergy-Pontoise (examinatrice)

J'ai aussi participé aux comités de recrutements suivants :

— Comité de séléction pour poste en CDD d'enseignant(e)-chercheur(e) à l'Université de Cergy-Pontoise pour la campagne 2023. Le profil du poste était en Systèmes, réseaux et sécurité.

— Comité de sélection pour un poste de maître de conférences à l'ENSEA, à Cergy-Pontoise, en section 27-61, pour la campagne 2022. Le profil du poste était en Sécurité des Réseaux et des Systèmes Communicants à faible empreinte.

1.4 Activités d'enseignement

Pendant ma thèse, j'ai effectué une mission d'enseignement à l'école Polytechnique, à raison de 64 heures par an entre 2010 et 2013. Il s'agissait de cours, TP, et projets en traitement du son, dans le cadre d'une UE de 2ème année appellée MODEX (Module expérimental).

Dans cette partie, je décris essentiellement mes activités d'enseignement à l'IMT Atlantique, où je suis arrivée en 2015. A l'IMT Atlantique, depuis 2018, l'enseignement est organisé de la manière suivante. En première année, les étudiants suivent un parcours de tronc commun, avec des enseignements scientifiques de base en mathématiques, en physique, et en informatique. En deuxième et en troisième année, chaque étudiant choisit une Thématique d'Approfondissement (TAF), qui vont définir un ensemble d'Unités d'Enseignement (UE) obligatoires (UE coeurs) et optionnelles (UE électives) à suivre sur une année complète. A noter que la formation était organisée très différement avant 2018. La fusion entre Télécom Bretagne et les Mines de Nantes a été l'occasion d'une réforme complète de l'enseignement, à laquelle j'ai eu la chance de participer.

Dans la configuration actuelle, en première année, j'interviens principalement sur les enseignements en mathématiques, en probabilités et statistiques, et en méthodes numériques. En deuxième et troisième année, j'interviens principalement dans la TAF "Mathematical and Computational Engineering" (MCE), dédiée au traitement de signal, au Machine Learning, et aux méthodes numériques, ainsi que dans la TAF "Information et Systèmes de Communications" (ISC), dédiée à la conception de systèmes de communications avec des composantes en codage source et en codage canal.

Comme dans mes activités de recherche, mes activités d'enseignement sont liées au codage source/canal, au traitement du signal, et au Machine Learning, avec aussi des thématiques plus génériques en probabilités, statistiques, et méthodes numériques.

1.4.1 Responsabilités en enseignement

— Depuis septembre 2020, je suis co-responsable avec Dominique Pastor de **l'UE coeur "Introduction to the theory and practice of Machine Learning"** (40 heures) de la TAF MCE. L'UE est suivie par environ 60 étudiants, et mobilise une dizaine d'enseignants-chercheurs et doctorants, intervenant chacun sur leurs sujets de prédilection.

— Depuis septembre 2019, je suis responsable de **l'UE optionnelle "Compression de données : du codage de sources à la réalité virtuelle"** (40 heures) de la TAF ISC. L'UE est suivie par une dizaine d'étudiants chaque année. Je suis l'intervenante principale, et trois autres enseignants-chercheurs interviennent sur des parties spécifiques de l'UE.

— Nous avons mise en place un **parcours "développement informatique"** au sein de la TAF MCE, avec Lucas Drumetz et Pierre Tandeo. L'idée de ce parcours, qui n'existe que dans la TAF MCE, est de compléter la formation des étudiants avec une composante informatique liée aux thématiques de la TAF. Ce parcours se concrétise sous la forme de trois projets, chacun adossés à une UE coeur de la TAF. Un premier projet adresse le calcul parallèle pour des problèmes d'optimisation (UE Méthodes Numériques). Un deuxième projet concerne l'estimation de moments sur de grosses bases de données (UE processus stochastiques). Le troisième projet s'intéresse aux bonnes pratiques de programmation et au travail collaboratif avec Git, pour des problèmes de Machine Learning (UE Machine Learning).

1.4.2 Cours actuels

Dans cette partie, je fais la liste des cours dans lesquels j'interviens actuellement.

Depuis 2018, dans le parcours commun de formation (1ère année) :

— *Calcul Scientifique et applications* : TD, TP (10 heures par an). Sujets abordés : systèmes linéaires, systèmes non-linéaires, optimisation

— *Probabilités et Statistiques* : cours, TD, TP (30 heures par an). Sujets abordés sur la partie probabilités : lois usuelles, variables aléatoires absolument continues, moments, couples et vecteurs aléatoires, vecteurs Gaussiens, convergence de suites de variables aléatoires. Sujets abordés pour la partie statistiques : estimation, tests d'hypothèses, régression linéaire

Depuis 2015, dans le parcours Ingénieurs en Apprentissage (1ère année) :

— *Probabilités* : TD et TP (15 heures par an). Sujets abordés : variables aléatoires discrètes et continues, lois usuelles, moments, variables aléatoires Gausiennes, couples et vecteurs aléatoires

Depuis 2019, dans la TAF MCE (2ème et 3ème année) :

— *Machine Learning* (UE coeur) : cours et encadrement de TP (15 heures par an). Sujets abordés : clustering, test d'hypothèses, SVM, réseaux de neurones, ACP

— *Méthodes numériques* (UE coeur) : TD et TP (15 heures par an), Sujets abordés : optimisation sans contraintes, optimisation sous contraintes, optimisation stochastique, décomposition en valeurs singulières

Depuis 2019, dans la TAF ISC (2ème et 3ème année) :

— *UE Codage canal* (UE optionnelle) : cours et encadrement de TP (6 heures par an). Sujets abordés : codes et décodeurs LDPC

— *UE Codage sources* (UE optionnelle) : cours, TD, TP, projet (30 heures par an), Sujets abordés : méthodes de base en codage de sources (codage entropique, quantification, codage par transformées, codage prédictif, etc.), standards de compression d'images et vidéo

1.4.3 Cours dans l'ancienne version de la formation

Dans cette partie, je fais la liste des cours principaux dans lesquels je suis intervenue dans l'ancienne version du parcours de formation à Télécom Bretagne. Ces cours n'existent

plus sous la même forme dans la formation actuelle, même si une partie de leur contenu à souvent été repris dans des UE construites différemment.

En 2016, en Master Of Science (1ère année) :

— *Algèbre linéaire* : cours et TD (10 heures par an). Sujets abordés : espaces vectoriels, résolution de systèmes linéaires, pivot de Gauss, inversion de matrices, transformations linéaires, déterminants, valeurs propres, vecteurs propres, produit scalaire

Entre 2015 et 2016 dans le parcours commun de formation (1ère année)

— *Probabilités* : cours, TD, TP (15 heures par an). Sujets abordés : lois usuelles, variables aléatoires absolument continues, moments, couples et vecteurs aléatoires, vecteurs Gaussiens, convergence de suites de variables aléatoires.

Entre 2015 et 2018 dans le domaine Mathematiques et Traitement de Signal (1ère et 2ème année)

— *Processus Stochastiques* : TD et TP (15 heures par an). Sujets abordés : processus stationnaires, processus de Wiener, processus de Poisson, chaînes de Markov

— *Codage source* : cours TD (10 heures par an). Sujets abordés : codage entropique, quantification, codage par transformées, codage prédictif

— *Codage canal* : cours, TD, TP (15 heures par an). Sujets abordés : codes linéaires en bloc, codes convolutifs

1.4.4 Encadrement de projets

De mon point de vue, les projets constituent une partie importante de la formation des étudiants ingénieurs. Ils interviennent de manière complémentaire et transverse aux enseignements classiques. Ils permettent de creuser un sujet technique, mais aussi d'observer et de développer d'autres types de compétences chez les étudiants (gestion de projets, autonomie, travail en équipe, synthèse et présentation des résultats, etc.). Je propose et encadre donc régulièrement des projets à différents stades de la formation.

Projets de développement en première année, groupes de 3 ou 4 étudiants

— En 2023, projet “Probabilités pour le tirage au sort de la Ligue des Champions” (calcul de probabilités)

— En 2021, projet “Décodeur LDPC neuronal” (codes LDPC, réseaux de neurones), avec Raphaël Le Bidan

— En 2020, projet “Implémentation d'un système de cryptanalyse” (cryptographie, cryptanalyse, attaques par corrélation, codes correcteurs d'erreurs)

— En 2019, projet “Conception d’un système de vote électronique” (cryptographie)

— En 2018, projet “Comprendre et implémenter les mécanismes de base du BitCoin” (cryptographie, fonctions de hachage, blockchain)

— En 2017, projet “Construction automatique de plans de pièces par mesures acoustiques” (traitement du signal, réponse impulsionnelle, mesures acoustiques), avec Jean-Marc Autret

Projets de recherche en troisième année, un seul étudiant

— Année 2019/2020, projet “Etude et implementation d’une chaine de communication de signaux AIS sur plate-forme radio logicielle” (systèmes de communications, radio logicielle), avec Thierry Le Gall et Christophe Laot

— Année 2017/2018, projet “Algorithmes de clustering distribués” (clustering, Mean-Shift)

— Année 2017/2018, projet “Projection de graphes” (théorie des graphes, embedding de graphes, traitement de signal sur graphes), avec Vincent Gripon

— Année 2016/2017, projet “Apprentissage décentralisé par protocole Gossip” (clustering, algorithm EM, protocole Gossip)

— Année 2016/2017, projet “Clustering distribué sur données compressées” (clustering, compression distribuée, réseaux de capteurs), avec Dominique Pastor

Projet de développement en troisième année, formation d’ingénieurs en apprentissage, groupe de 2 étudiants

— En 2019, projet “Implémentation d’un codeur arithmétique pour le codage de sources” (codage de sources, codage arithmétique), avec Christophe Laot

Projets dans la TAF MCE (2ème et 3ème année)

— Dans le cadre du parcours développement de la TAF MCE mis en place depuis 2019, j’encadre chaque année le projet intitulé “Machine Learning et bonnes pratiques de programmation”. Tous les étudiants de la TAF (environ 60 étudiants) réalisent ce projet, par groupe de 3 ou 4 étudiants. L’objectif du projet est de développer un seul pipeline en Python pour tester la classification sur deux jeux de données différents. Ils doivent implémenter leur solution en Python, en utilisant Git pour le versionnement et le travail collaboratif sur le code. En plus des résultats techniques, le rapport final doit intégrer une réflexion sur les bonnes pratiques en matière de programmation.

— Dans le cadre de l'UE optionnelle "projets de développement en Machine Learning" de la TAF MCE, j'ai proposé en 2020 un projet intitulé "Fault-tolérant Neural Networks", pour un groupe de 2 étudiantes.

1.4.5 Tutorat

Tutorat d'étudiants en formation par apprentissage : j'ai été la tutrice de deux étudiants en formation d'ingénieur par apprentissage à l'IMT Atlantique :

— *Elouan Le Duc*, entre 2019 et 2022, à Orange Labs, à Lannion, sur le sujet "Ingénierie pour l'internet des objets"

— *Yassine Habib*, entre 2016 et 2019, à Thalès, à Palaiseau, sur le sujet "Détection et suivi de personnes et de véhicules depuis un drone". Yassine a reçu le prix du meilleur apprentissage de l'IMT en 2020.

A ce titre, j'ai eu l'occasion de suivre et d'accompagner la montée en compétence de ces étudiants pendant les trois années de leur formation. J'ai eu des réunions régulières avec chacun d'entre eux à l'occasion de leurs périodes à l'école. Je me suis rendue à deux ou trois reprises dans leurs entreprises pour discuter avec eux et avec leurs encadrants de leur travail.

Tutorat d'étudiants en stage de fin d'études : Les étudiants en troisième année à l'IMT Atlantique effectuent leur stage de fin d'étude, et sont suivis par un tuteur ou une tutrice côté école. Il s'agit de discuter régulièrement avec l'étudiant et son encadrant, pour s'assurer que tout se passe bien, d'apporter un soutien vis à vis des livrables du stage, d'évaluer le rapport de stage, et de participer à la soutenance. J'ai été tutrice des étudiants suivants :

— En 2022, Christopher Jabea, stage "Analyse d'images" à Thalès

— En 2021, Maher Ouali, stage "Data Science : recommandation d'items sur un menu" à MIAP

— En 2020, Baptiste Gueuziec, stage "Choix de capteur pour planification de mission", à Thalès

— En 2019, Maya Assal, stage "Détection des fraudes sur Orange Money", à Orange

— En 2019, Ricardo Andreassen, stage "ML-based cycle prediction", à Harmonic

— En 2017, Antoine Houssais, stage "Assistant chef de produit" à SagemCom

Projet Découverte en première année à Télécom Bretagne : Dans l'ancien parcours de formation à Télécom Bretagne, les élèves de première année suivaient un projet Découverte comportant un certain nombre d'activités en lien avec le travail en équipe, la gestion de projet, la préparation de présentations, etc. Entre 2015 et 2018, j'ai régulièrement encadré des groupes de 8 étudiants pour les activités suivantes :

— *Course d'orientation* : chaque groupe dispose de 30 minutes pour retrouver un certain nombre de balises dispersées sur le campus, après résolution d'énigmes. Le rôle de tutrice est de suivre le groupe, sans l'aider, pendant la course, puis d'effectuer ensuite un débriefing avec les étudiants. C'est l'occasion d'une réflexion et d'une discussion sur l'organisation du travail en équipe.

— *Pecha Kucha* : chaque étudiant du groupe doit préparer un Pecha Kucha (20 slides, 20 secondes par slide) sur un sujet de son choix, et le présenter pendant la séance. Les étudiants ont ensuite 1h30 pour préparer ensemble un nouveau Pecha Kucha sur le sujet : "comment faire une bonne présentation à l'oral". Il s'agit de travailler à la fois sur les présentations orales, et sur le travail en équipe.

1.4.6 Formation Continue

La Formation Continue (FC) est destinée à des salariés en entreprise. Je suis impliquée dans les deux FC suivantes :

— Je suis responsable et seule intervenante de la **FC "Techniques de compression : du codage de sources à la vidéo streaming"**. Il s'agit d'une formation sur trois jours, qui aborde les fondamentaux du codage de sources (codage entropique, quantification, codage par transformées, codage prédictif), les normes de compression multimédia (images et vidéo), la compression pour le streaming vidéo.

— Je participe à la **FC "Communications Numériques"**, dont le responsable est Sébastien Houcke. Il s'agit d'une formation sur 5 jours, qui présente l'ensemble des éléments de la chaîne de communications. Dans cette formation, j'interviens sur le cours "OFDM", et sur le TP "Simulation d'une chaîne de communications".

1.5 Publications

1.5.1 Articles de revues

1. Belaïd Hamoum, **Elsa Dupraz**, Channel model and decoder with memory for DNA data storage with nanopore sequencing, *IEEE Access*, vol. 11, pp. 52075-52087, May 2023
2. Alireza Tasdighi, **Elsa Dupraz**, An end-to-end scheme for learning over compressed data transmitted through a noisy channel, in *IEEE Access*, vol. 11, pp. 8254 – 8267, January 2023
3. Khaled Alhaj Ali, Amer Baghdadi, **Elsa Dupraz**, Mathieu Léonardon, Mostafa Rizk and Jean-Philippe Diguet, MOL-based In-Memory Computing of Binary Neural Networks, accepted at *IEEE Transactions on Very Large Scale Integration Systems*, March 2022
4. Jonathan Kern, **Elsa Dupraz**, Abdeldjalil Aïssa-El-Bey, Lav R. Varshney, and François Leduc-Primeau, Optimizing the Energy Efficiency of Unreliable Memories for Quantized Kalman Filtering, *Special issue "Machine Learning, Signal, and/or Image Processing Methods to Enhance Environmental Sensors" of MDPI journal Sensors*, vol. 22, no 3, p. 853, 2022
5. **Elsa Dupraz**, Mohamed Yaoumi, Self-Corrected Belief-Propagation decoder for source coding with unknown source statistics, *IEEE Communication Letters*, vol. 25, no 7, pp. 2133-2137, 2021
6. Fangping Ye, Navid Mahmoudian Bidgoli, **Elsa Dupraz**, Aline Roumy, Karine Amis, Thomas Maugey, Bit-plane coding in extractable source coding : optimality, modeling, and application to 360° data, *IEEE Communication Letters*, vol. 25, no 5, pp. 1412-1416, 2021
7. **Elsa Dupraz**, François Leduc-Primeau, Noisy Density Evolution With asymmetric deviation models, *IEEE Transactions on Communications*, vol. 69, no 3, pp. 1403-1416, 2020
8. Mohamed Yaoumi, **Elsa Dupraz**, François Leduc-Primeau, Frederic Guilloud, Energy optimization of quantized Min-Sum decoders for protograph-based LDPC codes, *Annals of Telecommunications*, vol. 75, no 11, p. 615-621, 2020
9. Mai Quyen Pham, Aline Roumy, Thomas Maugey, **Elsa Dupraz**, Michel Kieffer,

Optimal reference selection for random access in predictive coding schemes, *IEEE Transactions on Communications*, vol. 68, no 9, pp. 5819-5833, 2020

10. Thomas Maugey, Aline Roumy, **Elsa Dupraz**, Michel Kieffer, Incremental coding for extractable compression in the context of Massive Random Access, *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 251-260, March 2020
11. Marwa Ben Abdessalem, Amin Zribi, Tadashi Matsumoto, **Elsa Dupraz**, Ammar Bouallegue, LDPC-based Joint Source Channel Coding and Decoding Strategies for single relay cooperative communications, *Elsevier Physical Communications*, vol. 38, February 2020
12. **Elsa Dupraz**, Aline Roumy, Thomas Maugey, Michel Kieffer, Rate-Storage Regions for Extractable Source Coding with Side Information, *Elsevier Physical Communications*, vol. 37, December 2019
13. Fangping Ye, **Elsa Dupraz**, Zeina Mheich, Karine Amis, Optimized Rate-Adaptive Protograph-Based LDPC Codes for Source Coding with Side Information, *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 3879-3889, June 2019
14. **Elsa Dupraz**, David Declercq, Bane Vasic, Asymptotic Error Probability of the Gallager B Decoder under Timing Errors, *IEEE Communication Letters*, vol. 21, no 4, p. 698-701. January 2017
15. **Elsa Dupraz**, David Declercq, Bane Vasic, Valentin Savin, Analysis and Design of Finite Alphabet Iterative Decoders Robust to Faulty Hardware, *IEEE Transactions on Communications*, vol.63, no 8, pp.2797 - 2809 June 2015
16. Christiane L. Kameni Ngassa, Valentin Savin, **Elsa Dupraz**, David Declercq, Density Evolution and Functional Threshold for the Noisy Min-Sum Decoder, *IEEE Transactions on Communications*, vol.63, no 5, pp.1497 - 1509, May 2015
17. **Elsa Dupraz**, Valentin Savin, Michel Kieffer, Density Evolution for the Design of Non-Binary Low Density Parity Check Codes for Slepian-Wolf Coding, *IEEE Transactions on Communications*, vol.63, no 1, pp.25–36, January 2015
18. Francesca Bassi, Aurelia Fraysse, **Elsa Dupraz**, Michel Kieffer, *Rate-distortion bounds for Wyner-Ziv coding with Gaussian scale mixture correlation noise*, *IEEE Transactions on Information Theory*, vol. 30, no 12, pp. 7540–7546, October 2014
19. **Elsa Dupraz**, Aline Roumy, Michel Kieffer, Source coding with side information

at the decoder and uncertain knowledge of the correlation, *IEEE Transactions on Communications*, vol. 62, no 1, pp. 269–279, January 2014

1.5.2 Articles de conférences internationales

1. Ismaila Salihou Adamou, **Elsa Dupraz**, Amin Zribi, Tad Matsumoto, Error-Exponent of Distributed Hypothesis Testing for Gilbert-Elliot Source Models, accepted at *International Symposium on Topics in Coding* (ISTC) 2023
2. Jeremy Nadal, Mohamed Yaoumi, **Elsa Dupraz**, Frederic Guilloud, François Leduc-Primeau, Energy Optimization of Faulty Quantized Min-Sum LDPC Decoders, accepted at *International Symposium on Topics in Coding* (ISTC) 2023
3. Jiahui Wei, **Elsa Dupraz**, Philippe Mary, Asymptotic and non-asymptotic rate-loss bounds for linear regression with side information, accepted at the *31st European Signal Processing Conference* (EUSIPCO), September 2023
4. Belaïd Hamoum, Aref Ezzeddine, **Elsa Dupraz**, Synchronization algorithms from high-rate LDPC codes for DNA data storage, accepted at the *International Conference on Digital Signal Processing* (DSP), June 2023
5. Jonathan Kern, Sébastien Henwood, Gonçalo Mordido, **Elsa Dupraz**, Abdeldjalil Aïssa-El-Bey, Yvon Savaria, and François Leduc-Primeau, MemSE : Fast MSE Prediction for Noisy Memristor-Based DNN Accelerators, *IEEE International Conference on artificial intelligent circuits and systems* (AICAS), June 2022
6. Belaid Hamoum, **Elsa Dupraz**, Laura Conde-Canencia, A DNA Data Storage Channel Model Trained on Genomic Data with Nanopore Sequencing, *1st International Conference on Data Storage in Molecular Media* (DSMM), February 2022
7. **Elsa Dupraz**, Lav R. Varshney, and François Leduc-Primeau, Power-Efficient Deep Neural Networks with Noisy Memristor Implementation, *Information Theory Workshop* (ITW), Kanazawa, Japan, 2021
8. Adomas Baliuka, **Elsa Dupraz**, Harald Weinfurter, Open Source LDPC-based error correction, accepted for poster presentation at QCrypt 2021, Amsterdam, The Netherlands, August 2021
9. Jérémy Nadal, Simon Brown, **Elsa Dupraz**, and François Leduc-Primeau, Towards an Accurate High-Level Energy Modelfor LDPC Decoders, *International Symposium on Topics in Coding* (ISTC), Montreal, Canada, September 2021, Invited paper

10. Belaid Hamoum, **Elsa Dupraz**, Laura Conde-Canencia, Dominique Lavenier, Channel Model with Memory for DNA Data Storage with Nanopore Sequencing, *International Symposium on Topics in Coding* (ISTC), Montreal, Canada, September 2021
11. Jonathan Kern, **Elsa Dupraz**, Abdeldjalil Aïssa-El-Bey, François Leduc-Primeau, Improving the Energy-Efficiency of a Kalman Filter using Unreliable Memories, *International Conference on Acoustic, Speech, and Signal Processing* (ICASSP), June 2021
12. Jeremy Nadal, Mickael Fiorentino, **Elsa Dupraz**, Francois Leduc-Primeau, A Deeply Pipelined, Highly Parallel and Flexible LDPC Decoder, *IEEE International Newcas conference*, Montreal, Canada, June 2020
13. **Elsa Dupraz**, Lav R. Varshney, Noisy In-Memory Recursive Computation with Memristor Crossbars, *International Symposium on Information Theory* (ISIT), Los Angeles, USA, June 2020
14. **Elsa Dupraz**, Lav R. Varshney, Energy-Efficient Machine Learning Algorithms, *Conference on Information Theory and Complex Systems* (TINKOS), Belgrade, Serbia, October 2019
15. Mohamed Yaoumi, **Elsa Dupraz**, Francois Leduc-Primeau, Frederic Guilloud, Energy-Efficient Protograph-Based LDPC codes, *Conference on Information Theory and Complex Systems* (TINKOS), Belgrade, Serbia, October 2019
16. Mohamed Yaoumi, François Leduc-Primeau, **Elsa Dupraz**, Frederic Guilloud, Optimization of Protograph LDPC Codes based on High-Level Energy Models, accepted at *16th International Symposium on Wireless Communication Systems* (ISWCS), Oulu, Finland, August 2019
17. **Elsa Dupraz**, Lav R. Varshney, Binary Recursive Estimation on Noisy Hardware, accepted at *International Symposium on Information Theory* (ISIT), Paris, France, July 2019
18. **Elsa Dupraz**, François Leduc-Primeau, François Gagnon, High-Throughput LDPC Decoding Achieved by Code and Architecture Co-Design, *International Symposium on Turbo Codes and Iterative Information Processing* (ISTC), Hong Kong, December 2018, Invited Paper
19. Nicolas Grelier, Carlos Eduardo Rosar Kos Lassance, **Elsa Dupraz**, Vincent Gripon, Graph-Projected Signal Processing, *IEEE International Conference on Signal*

and Information Processing (GlobalSIP), Anaheim, USA, November 2018

20. Fangping Ye, Zeina Mheich, **Elsa Dupraz**, Karine Amis, Optimized Short-Length Rate-Adaptive LDPC Codes for Slepian-Wolf Source Coding, *International Conference on Telecommunication* (ICT), Saint-Malo, France, June 2018

21. Mael Bompais, Hamza Ameur, Dominique Pastor, **Elsa Dupraz**, The p-value as a New Similarity Function for Spectral Clustering in Sensor Networks, *Statistical Signal Processing Workshop* (SSP), Freiburg, Germany, June 2018

22. Nicolas Grelier, Carlos Eduardo Rosar Kos Lassance, **Elsa Dupraz**, Vincent Gripon, Graph-Projected Signal Processing, *Graph Signal Processing Workshop* (GSP), Lausanne, Switzerland, June 2018

23. **Elsa Dupraz**, Dominique Pastor, Decentralized clustering algorithm over compressed data, *Conference on Information Theory and Complex Systems* (TINKOS), Belgrade, Serbia, June 2018

24. Fangping Ye, **Elsa Dupraz**, Karine Amis, Rate-adaptive LDPC code construction for Free-Viewpoint Television, *Conference on Information Theory and Complex Systems* (TINKOS), Belgrade, Serbia, June 2018

25. **Elsa Dupraz**, Dominique Pastor, François-Xavier Socheleau, A Statistical Signal Processing Approach to Clustering over Compressed Data, *International Conference on Acoustics, Speech and Signal Processing* (ICASSP), Calgary, Canada, April 2018

26. Zeina Mheich, **Elsa Dupraz**, Short Length Non-binary Rate-Adaptive LDPC Codes for Slepian-Wolf Source Coding, *Wireless Communications and Networking Conference* (WCNC), Barcelona, Spain, April 2018

27. **Elsa Dupraz**, K-means Algorithm over Compressed Binary Data, *Data Compression Conference* (DCC), Utah, United States, March 2018

28. **Elsa Dupraz**, Thomas Maugey, Aline Roumy, Michel Kieffer, Rate-Distortion Performance of Sequential Massive Random Access to Gaussian Sources with Memory, *Data Compression Conference* (DCC), Utah, United States, March 2018

29. Velimir Ilić, **Elsa Dupraz**, Bane Vasic, Generic Architectures for Uniformly Reweighted APP Decoders, *International Conference on Advanced Technologies, Systems, and Services in Telecommunications* (TELSIKS), Nis, Serbia, October 2017, Invited Paper

30. **Elsa Dupraz**, Bane Vasic, David Declercq, Performance of Taylor-Kuznetsov memories under timing errors, *International Conference on Communications* (ICC), Paris, France, May 2017
31. **Elsa Dupraz**, Distributed K-means over Compressed Binary Data, *National Conference on Information Theory and Complex Systems* (TINKOS), Belgrade, Serbia, October 2016
32. Satish Kumar Grandhi, **Elsa Dupraz**, Christian Spagnol, Valentin Savin, Emanuel Popovici, CPE : Codeword Prediction Encoder, *European Test Symposium*, Amsterdam, Netherlands, May 2016
33. **Elsa Dupraz**, Valentin Savin, Satish Kumar Grandhi, Emanuel Popovici, David Declercq, Practical LDPC Encoders Robust to Hardware Noise, *International Conference on Communications* (ICC), Kuala Lumpur, Malaysia, May 2016
34. **Elsa Dupraz**, David Declercq, Evaluation of the Robustness of LDPC Encoders to Hardware Noise, *BlackSeaCom*, 2015, Invited Paper
35. **Elsa Dupraz**, D. Declercq, B. Vasic, Analysis of Taylor-Kuznetsov Memory using One-Step Majority Logic Decoder, *Information Theory and Applications Workshop* (ITA), 2015, Invited paper
36. Velimir Ilic, **Elsa Dupraz**, David Declercq, Bane Vasic, Uniformly reweighted APP Decoder for memory efficient decoding of LDPC Codes, *Allerton*, 2014
37. **Elsa Dupraz**, David Declercq, Bane Vasic, Valentin Savin, Finite Alphabet Iterative Decoders Robust to Faulty Hardware : Analysis and Selection, *International Symposium on Turbo Codes and Iterative Information Processing*, 2014
38. Velimir Ilic, **Elsa Dupraz**, David Declercq, Bane Vasic, On the Memory Complexity of APP Decoders for LDPC Codes, *ICT Forum* 2014, Serbia, Invited Paper
39. Velimir Ilic, **Elsa Dupraz**, David Declercq, Bane Vasic, Memory Efficient APP Decoding of LDPC Codes, *National Conference on Information Theory and Complex Systems* 2014, Serbia
40. **Elsa Dupraz**, Aline Roumy, Michel Kieffer, Universal Wyner-Ziv coding for Gaussian sources, *International Conference on Acoustic, Speech, and Signal Processing* (ICASSP), 2013
41. **Elsa Dupraz**, Aline Roumy, Michel Kieffer, Practical coding scheme for universal source coding with side information at the decoder, *Data Compression Conference* (DCC), 2013

42. **Elsa Dupraz**, Aline Roumy, Michel Kieffer, Source coding with side information at the decoder : Models with uncertainty, performance bounds, and practical coding schemes., *International Symposium on Information Theory and its Applications* (ISITA), 2012
43. **Elsa Dupraz**, Francesca Bassi, Thomas Rodet, Michel Kieffer, Distributed coding of sources with bursty correlation, *International Conference on Acoustic, Speech, and Signal Processing* (ICASSP), 2012
44. **Elsa Dupraz**, Gael Richard, Robust frequency-based audio fingerprinting, *International Conference on Acoustic, Speech, and Signal Processing* (ICASSP), 2010

1.5.3 Articles de conférences nationales

1. Ismaila Salihou-Adamou, **Elsa Dupraz**, Tad Matsumoto, Test d'hypothèses distribué pour des modèles de sources générales non-iid, non-stationnaires, et non-ergodiques, accepted at *GRETSI* 2023
2. Jiahui Wei, **Elsa Dupraz**, Philippe Mary, Régions atteignables pour la régression linéaire sur données compressées avec information adjacente, accepted at *GRETSI* 2023
3. Fangping Ye, **Elsa Dupraz**, Zeina Mheich, Karine Amis, Construction de Codes LDPC Compatibles en Rendement pour le Codage de Sources avec Information Adjacente, *Actes du GRETSI* 2019
4. Mohamed Yaoumi, **Elsa Dupraz**, François Leduc-Primeau, Frederic Guilloud, Optimisation de la Consommation d'Energie pour des Codes LDPC Construits à Partir de Protographes, *Actes du GRETSI*, 2019
5. **Elsa Dupraz**, David Declercq, Bane Vasic, Stabilité des Mémoires de Taylor-Kuznetsov construites à partir d'un Décodeur LDPC de type Gallager B, *Actes du GRETSI* 2015
6. **Elsa Dupraz**, Aline Roumy, Michel Kieffer, Codage distribué dans des réseaux de capteurs avec connaissance incertaine des corrélations, *Actes du GRETSI* 2013
7. **Elsa Dupraz**, Aline Roumy, Michel Kieffer, Codage de sources avec information adjacente et connaissance imparfaite de la correlation : le problème des cadrans, *Actes du GRETSI* 2013

Chapitre 2

Introduction

2.1 Un outil central : le codage canal

Le codage canal est un sujet de recherche assez ancien, qui a trouvé une motivation certaine dans le résultat de Shannon de 1948 sur la capacité des canaux de communications [1]. Introduits en 1993, les Turbo Codes ont constitué la première famille de codes permettant d'atteindre une performance proche de la capacité pour un décodage relativement peu complexe [2]. Ils ont été suivis par les codes LDPC, décrits initialement dans la thèse de Gallager en 1968 [3] et re-découverts à la fin des années 90 par McKay [4], puis par les codes Polaires inventés par Arikan dans les années 2000 [5]. Pour chacune de ces familles de codes, de nombreux travaux ont proposé des outils d'analyse de performance, des méthodes de constructions de codes, des solutions pour améliorer ou accélérer le écodage, des implémentations matérielles efficaces, etc., ce qui a permis d'en faire un outil central du domaine des télécommunications.

Bien sûr, il existe encore de nombreuses questions ouvertes, comme la construction de codes courts performants, la diminution de la complexité et de la latence de décodage, ou l'optimisation de la consommation d'énergie des décodeurs. Cependant, mon objectif dans ce document est de démontrer que les codes correcteurs d'erreurs dépassent largement le cadre des télécommunications, et présentent une grande utilité dans de nombreux domaines connexes, comme le codage de sources distribué [6], le stockage de données sur différents supports, ou encore le calcul en mémoire [7]. Dans ces applications, les erreurs ne sont plus introduites par un canal de communication "standard", de type Gaussien ou à évanouissement, mais par un support de stockage, un circuit de calcul défaillant, ou encore un modèle de corrélation entre des sources.

La plupart de ces applications ne constituent pas une simple utilisation de codes correcteurs d'erreurs existants "sur l'étagère", mais introduisent en réalité des nouveaux problèmes de recherche à part entière : parce qu'elles induisent de nouveaux modèles

d'erreurs (erreurs d'insertions et de deletions pour le stockage de données dans des molécules d'ADN), parce que le critère de performance est différent (communications "goal-oriented"), ou tout simplement parce qu'elles nécessitent de modifier la structure du code en fonction de l'objectif (codage de sources, calcul bruité).

2.2 Vers d'autres applications du codage de canal

Je présente maintenant quelques unes de ces applications sur lesquelles j'ai travaillé ces dernières années. Pour étudier ces applications, il est nécessaire d'utiliser des méthodes provenant de différents domaines, tels que la théorie de l'information, le codage de canal, le codage de sources, et le machine learning.

2.2.1 Codage de sources pour les communications interactives

La télévision interactive constitue un paradigme récent de visualisation de vidéos dans lequel les utilisateurs ont la possibilité de naviguer librement dans différentes vues d'une même scène [8]. On peut penser par exemple à un match de basket, où les utilisateurs pourraient choisir de regarder la vue près d'un panier, ou au contraire depuis la ligne médiane. Les vues présentent des dépendances statistiques les unes avec les autres, ce qu'un schéma de compression efficace devrait pouvoir exploiter. Dans la littérature, les premières solutions proposées pour cette application consistaient soit à coder conjointement l'ensemble des vues [9], et donc à toutes les transmettre à l'utilisateur, soit à coder tous les sous-ensembles possibles de vues [10], au prix d'un coût important en stockage.

Dans l'idée de développer des solutions plus efficaces, nous avons tout d'abord proposé une modélisation générale de ce problème, pour pouvoir ensuite l'étudier du point de vue de la théorie de l'information. Dans notre modélisation, on considère un ensemble de sources corrélées et on suppose que chaque utilisateur veut accéder à un sous-ensemble, inconnu *a priori*, de ces sources. On défini ensuite deux types de débits différentes : un débit pour le stockage des sources sur le serveur, et un débit moyen de transmission des sous-ensemble de sources aux utilisateurs. Cette modélisation est désignée sous le nom de "communications interactives". Elle représente non seulement le cas de la télévision interactive, mais permet aussi de traiter d'autres applications comme la navigation dans des images à 360°.

Nous avons ensuite proposé une analyse de théorie de l'information des communi-

cations interactives, pour déterminer les limites fondamentales de ces systèmes, et en particulier comprendre le compromis entre le débit de stockage et les débits de transmission. Cette analyse a permis de montrer que pour atteindre les débits minimum, il est indispensable d'utiliser des constructions incrémentales des mots de code, de manière à extraire uniquement l'information nécessaire pour servir les requêtes de l'utilisateur. Suite à cela, nous avons proposé des schémas de codage pratiques pour les communications interactives, en utilisant les enseignements de l'analyse théorique. Ces schémas ont été construits à partir de codes incrémentaux basés sur des codes LDPC compatibles en rendement. Il n'a pas été possible d'utiliser directement les codes existants pour du codage de canal, et nous avons donc développé de nouvelles solutions compatibles en rendement adaptées au codage de sources. Nous avons enfin appliqué ces schémas de codage à des données réelles issues d'une application particulière : les images à 360° où les utilisateurs veulent accéder uniquement à certaines parties de l'image, et obtenu des performances intéressantes.

2.2.2 Calcul sur circuits bruités

La loi de Moore, énoncée en 1965, prédisait que le nombre de transistors sur les puces électroniques allait doubler tous les deux ans [11]. L'évolution du nombre de transistors a effectivement suivi cette prédiction empirique, et a permis des gains significatifs du point de vue de l'efficacité énergétique des puces. Mais les limites physiques des composants semblent avoir été atteintes, et il ne devrait plus y avoir d'évolution significative de la technologie CMOS dans les années à venir. C'est pourquoi le domaine du circuit étudie désormais d'autres alternatives pour obtenir des gains en énergie. L'une d'elles consiste à simplement diminuer drastiquement les tensions d'alimentations des circuits, ce qui réduira leur consommation d'énergie, au prix d'une introduction d'erreurs dans les processeurs et les mémoires implémentés sur ces circuits [12].

Une deuxième alternative part du constat que dans l'architecture usuelle de calcul dite de "Von Neumann", les processeurs sont sépares physiquement des mémoires. Les transferts de données conséquents entre ces deux parties du circuit sont responsables pour une bonne partie de sa consommation d'énergie [13]. L'alternative consisterait donc à effectuer une partie des opérations de calcul directement dans les mémoires [7], ce qui devrait être prochainement rendu possible par les nouvelles générations de mémoires non-volatiles, de types STT-MRAM, PCM, et ReRAM [14]. Ainsi, dans le cas d'un réseau de neurones, une unité de mémoire stockera l'ensemble des poids d'une couche linéaire,

et l'application de certains niveaux de tensions électriques en entrée de cette unité de mémoire permettra d'évaluer directement la sortie de la couche. Mais cela se ferra au prix d'une sensibilité accrue au bruit, dû en partie à la difficulté de programmer un grand nombre de niveaux différents dans les cellules mémoires.

Pour ces deux alternatives, il est donc nécessaire d'étudier la robustesse au bruit des algorithmes standards de traitement de signal et de machine learning. Si la robustesse de ces méthodes n'est pas suffisante, notamment en cas de bruit trop important, il sera indispensable d'y ajouter des mécanismes de correction d'erreurs pour les protéger du bruit, et les algorithmes de décodage devront eux-même être robustes au bruit.

Dans ce cadre, nous avons tout d'abord étudié les décodeurs LDPC implémentés sur circuit bruité. Nous avons proposé de nouveaux modèles d'erreurs représentant plus finement les effets du bruit (erreurs asymétriques, introduction de délais, etc.) dans de vrais circuits. Nous avons ensuite étudié théoriquement la robustesse au bruit des décodeurs LDPC, et proposé des stratégies d'optimisation permettant d'améliorer encore cette robustesse. Puis nous avons développé des méthodes permettant de caractériser la consommation d'énergie des décodeurs en fonction de la proportion de fautes introduites dans le circuit.

Dans un second temps, nous avons étudié des méthodes standards de traitement de signal et de machine learning implémentés sur circuit bruité : estimation binaire récursive, filtrage de Kalman, réseaux de neurones, etc. Pour chacune de ces méthodes, nous avons étudié de manière théorique l'effet des fautes sur la performance finale des algorithmes, ainsi que les gains potentiels en énergie. Une perspective importante de ces travaux consistera à implémenter des solutions de correction d'erreurs directement à l'intérieur de ces algorithmes. Il s'agit d'un problème difficile, car les opérations non-linéaires de ces algorithmes ne sont pas compatibles avec la structure linéaire des codes correcteurs d'erreurs usuels.

2.2.3 Stockage de données dans l'ADN

Le stockage de données sur des molécules d'ADN est une technologie émergente qui a pris de l'importance ces dernières années, notamment en raison de l'intérêt croissant d'entreprises comme Microsoft [15]. L'ADN constitue un support de stockage environ 1000 fois plus dense, et bien plus durable dans le temps, que les technologies classiques comme les disques dur ou les bandes magnétiques [16].

Une molécule d'ADN est constituée de deux brins symétriques, abritant chacun des

suites de bases de types A,C,G, et T. Une opération de synthèse chimique, utilisée couramment pour la fabrication de médicaments, permet de transformer une séquence numérique quaternaire en une molécule particulière. Cette molécule est ensuite dupliquée un grand nombre de fois à l'aide d'une réaction de type PCR. Ensuite, pour "lire" le contenu de la molécule, on utilise un séquenceur, qui va produire un grand nombre de copies bruitées des données stockées.

Cette technologie soulève de nouveaux défis du point de vue de la conception de codes correcteurs d'erreurs. En effet, l'opération de séquençage introduit un grand nombre d'erreurs dans les lectures : non seulement des substitutions, mais aussi des insertions et des déletions, que les codes usuels (LDPC, Turbo, etc.) ne savent pas bien traiter. Un enjeu important consiste aussi à exploiter les multiples copies bruitées des mêmes données, chaque copie contenant des réalisations d'erreurs différentes. Ce dernier enjeu est d'autant plus essentiel que la synthèse est actuellement un processus très onéreux, ce qui rend nécessaire l'utilisation de codes de rendements élevés. C'est pourquoi le problème de stockage de données dans l'ADN fait partie de mes perspectives.

2.2.4 Communications "goal-oriented"

Lorsque l'on conçoit un systèmes de télécommunications, on souhaite en général reconstruire les données à transmettre avec la meilleure qualité possible. Cependant, dans beaucoup d'applications, le but du récepteur n'est pas de reconstruire les données, mais de leur appliquer une tâche d'apprentissage : classification, reconnaissance et recommandation de contenu, etc. Il est donc nécessaire de revisiter les méthodes de conception de systèmes de télécommunications pour prendre en compte cet objectif. Ce paradigme est aujourd'hui étudié dans différents domaines, avec des noms différents : compression "task-oriented" en codage de sources [17], communications "goal-oriented" en théorie de l'information [18] et communications numériques [19], codage pour machines dans le domaine du deep learning [20].

Outre l'étude des performances atteignables par un système de codage dédié à l'apprentissage, une question fondamentale réside dans l'étude du compromis entre les tâches de reconstruction et d'apprentissage [21]. De plus, il est essentiel de développer des schémas de codage source et canal adressant ce compromis. Ces schémas devront permettre non seulement de reconstruire les données originales si nécessaire, mais aussi d'effectuer l'apprentissage sans avoir à reconstruire les données au préalable. C'est pourquoi ce sujet fait également partie de mes perspectives.

2.3 Démarche scientifique

Dans mes recherches, je tente le plus possible d'appliquer une démarche qui consiste à partir d'analyses théoriques des questions identifiées, pour aller ensuite vers des algorithmes et des implémentations pratiques.

2.3.1 Analyses théoriques

J'ai pour habitude de m'appuyer sur deux types d'analyses théoriques. Le premier type d'analyse utilise la théorie de l'information, qui permet de d'étudier les performances optimales atteignables par un système de codage, en fonction de ses objectifs (reconstruction des données, correction des erreurs, etc.). La caractérisation de la performance optimale permet bien entendu de comparer un schéma pratique à cette borne, pour savoir ce qu'il reste potentiellement à gagner en terme de performances. Mais elle présente aussi une autre utilité majeure : obtenir une compréhension du problème et des enseignements qui seront utiles pour la construction de schémas pratiques.

Par exemple, les preuves d'atteignabilité des schémas de codage de sources avec information adjacente, de type Slepian-Wolf [22] et Wyner-Ziv [23], utilisent des techniques de binning, dont on sait maintenant qu'elles peuvent être implémentées en pratique avec des codes correcteurs d'erreurs. En conséquence, lorsque l'on voit apparaître du binning dans les preuves pour du test d'hypothèse distribué [24], on peut se dire que les codes correcteurs d'erreur seront là aussi une bonne solution pratique.

Le deuxième type d'analyse sur lequel je m'appuie réside dans la prédiction théorique des performances d'algorithmes de correction d'erreur, de traitement de signal, ou d'apprentissage. Par exemple, pour les codes LDPC, des outils de type évolution de densité [25, 26] permettent d'exprimer la probabilité d'erreur du décodage, en fonction des paramètres du code et de l'algorithme utilisé. Ce type de méthode peut aussi être utilisé pour caractériser l'effet du bruit dans un algorithme particulier de traitement de signal ou d'apprentissage, et s'appuie le plus souvent sur des calculs de probabilités ou de moments de variables aléatoires. Outre l'utilité évidente en terme de caractérisation et d'optimisation des méthodes, il s'agit d'un outil puissant de vérification, puisque l'on peut comparer les résultats de simulations de Monte Carlo avec la prédiction théorique.

2.3.2 Schémas pratiques

J'attache aussi une certaine importance à ne pas me limiter à une analyse théorique, mais aussi à proposer des schémas de codage pratiques adaptés à l'application considérée. Cela nécessite un travail sur les contraintes spécifiques du domaine étudié, ainsi que le développement de modèles d'erreurs à la fois pertinents et utilisables dans nos méthodes de construction de codes. Par exemple, sur le sujet des erreurs introduites par le circuit, il est nécessaire de comprendre où et comment les erreurs sont introduites (dans les portes logiques, dans les mémoires, etc.), et quelles hypothèses sont raisonnables : indépendances statistiques, symétries, etc.

Pour travailler sur ces aspects, j'ai l'habitude de m'appuyer fortement sur des collaborations avec des experts du domaine, qui peuvent me guider dans l'analyse de leurs contraintes et difficultés spécifiques. Cela permet aussi de tester les solutions développées sur des données réelles, voire en conditions pratiques.

2.4 Structure du document

Ce rapport présente mes contributions sur les différents problèmes présentés dans cette introduction, en suivant la démarche décrite précédemment.

Dans le Chapitre 3, j'introduis les outils de base pour la construction et le décodage de codes LDPC, qui seront utilisés par la suite dans beaucoup de mes contributions. Dans le Chapitre 4, je présente le problème des communications interactives, les bornes théoriques et les schémas pratiques que nous avons développés, ainsi que les résultats obtenus sur des images à 360°. Dans le Chapitre 5, je décris le problème du décodage LDPC sur circuits bruités, les analyses théoriques de l'effet du bruit, et les méthodes d'évaluation de l'énergie que nous avons proposé. Dans le Chapitre 6, je présente les extensions aux méthodes d'apprentissage implémentées sur circuit bruités et dans des unités de calcul en mémoire. Enfin, je décris mes perspectives dans le Chapitre 7.

Chapitre 3

Codes LDPC

3.1 Introduction

Inventés par Gallager en 1963 [3] et redécouverts par MacKay en 1996 [27], les codes LDPC sont aujourd'hui présents dans de nombreux standards (DVB-S2, WIFI, 5G, etc.). Ils sont un outil central de mes recherches, ce qui justifie leur présentation détaillée dans ce chapitre.

3.2 Codes LDPC

Dans la suite, je présente la construction et le décodage des codes LDPC dans le contexte du codage canal. Dans ce contexte, on souhaite transmettre une séquence d'information notée $\mathbf{u}$, de longueur k. On tranforme tout d'abord cette séquence en un mot de code, désigné par $\mathbf{x}$ et de longueur n, que l'on transmet à travers un canal bruité produisant une sortie $\mathbf{y}$, de longueur n également. Le mot de code $\mathbf{x}$ prend ses valeurs dans l'alphabet $\{-1, 1\}$, tandis que la sortie $\mathbf{y}$ prend ses valeurs dans l'alphabet $\mathcal{Y}$, réel ou discret.

Le canal est caractérisé par une distribution de probabilité conditionnelle $P(y|x)$ et est supposé indépendant et identiquement distribué (i.i.d.). Les types de canaux les plus couramment utilisés sont le canal BSC (Binary Symmetric Channel) pour lequel $\mathcal{Y} = \{-1, 1\}$ et décrit par un paramètre p tel que

$$p = P(Y = 1|X = -1) = P(Y = -1|X = 1), \tag{3.1}$$

et le canal BiAWGN (Binary Input Additive White Gaussian Noise) avec $\mathcal{Y} = \mathbb{R}$, et décrit par une variance σ^2 tel que

$$(Y|X = x) \sim \mathcal{N}(x, \sigma^2). \tag{3.2}$$

En sortie du canal, on applique un décodeur LDPC qui utilise la structure du code pour reconstruire une version estimée $\hat{\mathbf{x}}$ du mot de code $\mathbf{x}$.

3.2.1 Matrice de parité et graphe de Tanner

Les codes LDPC sont des codes linéaires en blocs définis par une matrice de parité binaire notée H, de dimension $m \times n$, telle que tout mot de code $\mathbf{x}$ vérifie $H\mathbf{x} = 0$. On associe à H une matrice génératrice, notée G, de dimension $n \times (n - m)$, qui permet de réaliser l'encodage $\mathbf{x} = G\mathbf{u}$, et qui vérifie $HG = 0$. Si la matrice H est de rang plein, le rendement du code est donné par $R = 1 - m/n$.

La matrice de parité H d'un code LDPC est creuse, dans le sens où elle contient seulement un petit nombre de composantes non-nulles. Elle peut être représentée de manière équivalente par un graphe de Tanner, connectant n Noeuds Variables (NV) à m Noeuds Checks (NC). Le degré d_v du NV v indique le nombre de NC auxquels il est connecté, tandis que le degré d_c du NC c indique le nombre de NV auxquels il est connecté. On note $\mathcal{C}_v$ l'ensemble des NC connectés au NV v, et $\mathcal{V}_c$ l'ensemble des NV connectés au NC c.

Un code LDPC est dit régulier quand tous les NV ont le même degré d_v, et quand tous les NC ont le même degré d_c. Dans ce cas, le rendement du code est donné par $R = 1 - \frac{d_v}{d_c}$ [25]. Au contraire, un code LDPC est dit irrégulier quand le degré varie d'un noeud à l'autre. Dans ce cas, il existe deux représentations possibles, non équivalentes, des degrés du code : une représentation par distribution des degrés et une représentation par protographes.

3.2.2 Représentation par distribution des degrés

La distribution des degrés d'un code LDPC peut-être représentée sous la forme de deux polynômes $\lambda(x)$ et $\rho(x)$ ayant pour expressions [26, 28] :

$$\lambda(x) = \sum_{i=1}^{d_{v,\max}} \lambda_i x^{i-1}, \quad \rho(x) = \sum_{j=1}^{d_{c,\max}} \rho_j x^{j-1}. \tag{3.3}$$

Ici, $d_{v,\max}$ désigne le degré maximum des NV tandis que $d_{c,\max}$ fait référence au degré maximum des NC. Par ailleurs, λ_i représente la proportion d'arêtes qui sont connectées à des NV de degré i. De même, ρ_j désigne la proportion d'arêtes connectées à des NC de degré j. Les polynômes $\lambda(x)$ et $\rho(x)$ permettent d'exprimer le rendement du code sous la

forme compacte suivante [28] :

$$R = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - \frac{\sum_{j=1}^{d_{c,\max}} \rho_j/j}{\sum_{i=1}^{d_{v,\max}} \lambda_i/i}. \tag{3.4}$$

Les polynômes $\lambda(x)$ et $\rho(x)$ caractérisent une distribution des degrés dite "edge-perspective". De manière équivalente, on peut définir une distribution des degrés dite "node-perspective" à partir de deux polynômes $\tilde{\lambda}(x)$ et $\tilde{\rho}(x)$:

$$\tilde{\lambda}(x) = \sum_{i=1}^{d_{v,\max}} \tilde{\lambda}_i x^{i-1}, \quad \tilde{\rho}(x) = \sum_{j=1}^{d_{c,\max}} \tilde{\rho}_j x^{j-1}. \tag{3.5}$$

Cette fois, $\tilde{\lambda}_i$ (respectivement $\tilde{\rho}_j$) représente la proportion de NV de degré i, tandis que $\tilde{\rho}_j$ représente la proportion de NC de degré j. Le rendement du code s'écrit alors [28] :

$$R = 1 - \frac{\sum_{i=1}^{d_{v,\max}} i\tilde{\lambda}_i}{\sum_{j=1}^{d_{c,\max}} j\tilde{\rho}_j}. \tag{3.6}$$

De plus, les relations entre λ_i et $\tilde{\lambda}_i$ d'une part, et entre ρ_j et $\tilde{\rho}_j$ d'autre part, sont données par

$$\lambda_i = \frac{i\tilde{\lambda}_i}{\sum_{i=1}^{d_{v,\max}} i\tilde{\lambda}_i}, \quad \rho_j = \frac{j\tilde{\rho}_j}{\sum_{j=1}^{d_{c,\max}} j\tilde{\rho}_j}. \tag{3.7}$$

Ces deux distributions ont des usages différents : la distribution "node-perspective" est plus simple à utiliser dans des algorithmes de construction de codes à longueur finie, tandis que la distribution "edge-perspective" sera utile pour l'évaluation théorique des performances d'un code LDPC en fonction de sa distribution des degrés (méthode d'évolution de densité).

3.2.3 Représentation par protographes

Un protographe est une matrice B de dimension $m_c \times n_v$, où les coefficients de B sont des entiers positifs ou nuls [29, 30, 31]. Un protographe peut aussi être représenté sous la forme d'un graphe de Tanner entre n_v NV et m_c NC. Le coefficient $B_{i,j}$ indique le nombre de connections entre le NV à la position i et le NC à la position j. Un coefficient $B_{i,j} > 1$ indique qu'il y a $B_{i,j}$ arrêtes en parallèle entre le NV i et le NC j. Pour obtenir une matrice de parité H de dimension $m \times n$ à partir du protographe B, le graphe de Tanner

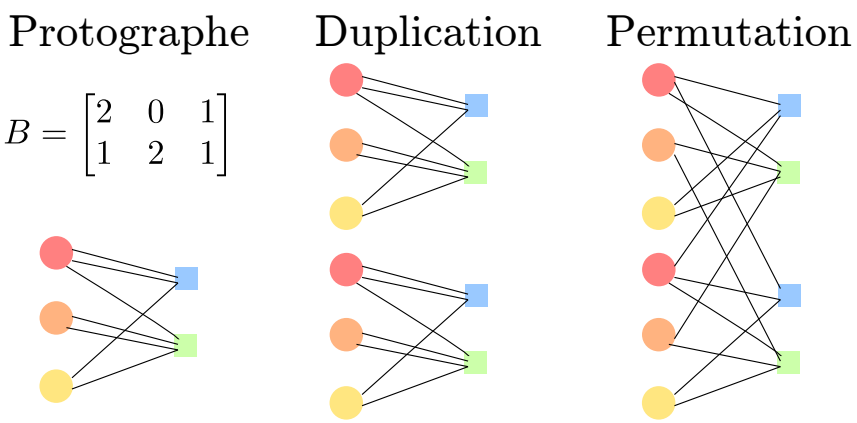

FIGURE 3.1 – Exemple d'une construction de matrice LDPC à partir d'un protographe. Les couleurs indiquent les types de noeuds contenus dans le protographe.

associé à B est copié Z fois, où $Z = n/n_v = m/m_c$. Puis les arrêtes sont permutées entre les Z composantes, de manière à obtenir un graphe de Tanner complètement connecté. Un exemple est représenté en Figure 3.1.

Une interprétation du protographe est la suivante : chaque coefficient $B_{i,j}$ de B indique le nombre de connections entre des NV de type i et des NC de type j. En copiant Z fois le protographe, on crée donc Z NV de type i et Z NC de type j qui se retrouveront dans la matrice H. Celle-ci est obtenue après une permutation des arrêtes, mais en respectant le nombre de connections indiqués dans B. Cette interprétation permet de comprendre pourquoi la représentation par protographe n'est pas équivalente à la représentation polynomiale. En effet, la représentation polynomiale n'inclut pas cette notion de "type" de noeud : les degrés des NV suivront la distribution polynomiale $\lambda(x)$, mais chaque NV pourra être connecté à n'importe quel NC, quelque soit son degré ou sa position dans le graphe. A l'inverse, une représentation par protographe impliquera qu'un certain NV soit connecté à exactement un certain nombre de NC d'un certain type.

Notons qu'une distribution de degrés $(\lambda(x), \rho(x))$, de même qu'un protographe B, définit un ensemble de codes qui partagent des propriétés similaires, notamment de performance, dont nous discuterons dans la suite. Avant cela, nous introduisons plusieurs solutions standards de décodage des codes LDPC.

3.3 Décodeurs LDPC

Je décris maintenant deux décodeurs LDPC parmi les plus utilisés : le décodeur Belief Propagation (BP) et le décodeur offset Min-Sum (MS) quantifié. Ces deux décodeurs sont des algorithmes de passage de messages [32] entre les NV et les NC du graphe de Tanner.

3.3.1 Décodeur BP

Dans le décodeur BP, chaque NV v calcule un message initial $u_v^{(0)}$ sous forme de logarithme de rapport de vraisemblance :

$$u_v^{(0)} = \log \frac{P(y_v|x_v = 0)}{P(y_v|x_v = 1)}, \tag{3.8}$$

où y_v représente la valeur reçue en sortie du canal. Ensuite, le décodeur fonctionne en L itérations. À l'itération $\ell \in [\![1, L]\!]$, pour tout $c \in \mathcal{C}_v$, le message du NC c vers le NV v est noté $u_{c\to v}^{(\ell)}$, et pour tout $v \in \mathcal{V}_c$, le message du NV v vers le NC c est noté $t_{v\to c}^{(\ell)}$. Les messages $u_{c\to v}^{(\ell)}$ et $t_{v\to c}^{(\ell)}$ sont calculés en utilisant les formules suivantes [25] :

$$u_{c\to v}^{(\ell)} = \frac{1 + \prod\limits_{v' \in \mathcal{V}_c \setminus v} \tanh\left(t_{v'\to c}^{(\ell-1)}\right)}{1 - \prod\limits_{v' \in \mathcal{V}_c \setminus v} \tanh\left(t_{v'\to c}^{(\ell-1)}\right)} \tag{3.9}$$

$$t_{v\to c}^{(\ell)} = u_v^{(0)} + \sum_{c' \in \mathcal{C}_v \setminus c} u_{c'\to v}^{(\ell)}. \tag{3.10}$$

Ensuite, à chaque itération, chaque NV calcule un message *a posteriori*, noté $t_v^{(\ell)}$, utilisant les d_v messages entrants et ayant pour expression :

$$t_v^{(\ell)} = u_v^{(0)} + \sum_{c \in \mathcal{C}_v} u_{c\to v}^{(\ell)}. \tag{3.11}$$

Le NV v prend ensuite une décision $\hat{x}_v^{(\ell)} = \text{sign}(t_v^{(\ell)})$ sur la valeur du symbole x_v dans le mot de code $\mathbf{x}$. Le décodage s'arrête lorsque la condition d'arrêt $H^T\hat{\mathbf{x}}^{(\ell)} = 0$ est vérifiée, ou lorsque les L itérations ont été réalisées.

3.3.2 Décodeur offset MS quantifié

Bien que présentant de très bonnes performances, le décodeur BP n'est pas très pertinent pour une implémentation, en raison de ses messages à valeurs réelles, et de son opération de calcul aux NC, qui utilise des tangentes hyperboliques complexes à implémenter. Une alternative consiste à utiliser un décodeur MS [33] quantifié, dans lequel l'opération de calcul aux NC sera approchée en utilisant un calcul de minimum. Le décodeur MS souffre d'une perte de performance par rapport au BP, mais il est possible d'introduire des paramètres supplémentaires dans les opérations de décodage, qui, une fois optimisés, permettront de rattraper une partie de l'écart avec le BP.

En gardant les mêmes notations pour les messages, les opérations effectuées dans le décodeur MS sont les suivantes. Les messages sont tout d'abord initialisés de la manière suivante :

$$u_v^{(0)} = \alpha \mathcal{Q}\left(\log \frac{P(y_v|X_v=0)}{P(y_v|X_v=1)}\right), \tag{3.12}$$

où α est un paramètre appelé le facteur d'échelle, et l'opérateur $\mathcal{Q}$ réalise la quantification uniforme du rapport de vraisemblance sur un nombre de bits q_s défini à l'avance. Ensuite, les messages aux NV et aux NC sont calculés en utilisant les formules suivantes :

$$u_{c\to v}^{(\ell)} = \left(\prod_{v'\in\mathcal{V}_c\setminus v} \operatorname{sgn}\left(t_{v'\to c}^{(\ell-1)}\right)\right) \max\left(\min_{v'\in\mathcal{V}_c\setminus v} |t_{v'\to c}^{(\ell-1)}| - \lambda, 0\right) \tag{3.13}$$

$$t_{v\to c}^{(\ell)} = \mathcal{Q}\left(u_v^{(0)} + \sum_{c'\in\mathcal{C}_v\setminus c} u_{c'\to v}^{(\ell)}\right). \tag{3.14}$$

Dans le calcul du message $u_{c\to v}^{(\ell)}$, l'opérateur sgn retourne le signe de $t_{v'\to c}^{(\ell-1)}$. Dans le calcul du message $t_{v\to c}^{(\ell)}$, l'opérateur $\mathcal{Q}$ sert uniquement à réaliser la saturation des messages au sein de l'intervalle de quantification. De plus, le paramètre λ est appelé offset [34]. L'optimisation des paramètres α et λ permet d'atteindre une performance proche du BP [35].

3.4 Evaluation de la performance d'un code LDPC par évolution de densité

L'évolution de densité [26, 25] permet de prédire la performance d'un décodeur LDPC pour un ensemble de codes de même distribution des degrés $(\lambda(x), \rho(x))$ ou de même pro-

tographe B. L'évolution de densité est à l'origine une méthode asymptotique qui considère que la longueur n du code tend vers l'infini. Dans cette partie, je décris tout d'abord cette méthode asymptotique, puis j'en présente une extension qui permet aussi de prédire la performance d'un ensemble de codes pour une longueur n finie.

3.4.1 Méthode asymptotique

L'évolution de densité consiste à calculer de manière récursive les distributions de probabilités des messages échangés dans le décodeur aux itérations successives $\ell \in [\![1, L]\!]$, ce qui permettra ensuite d'exprimer la probabilité d'erreur $P_e^{(\ell)}$ en sortie du décodeur.

Hypothèses pour réaliser l'évolution de densité

La méthode originale d'évolution de densité [26, 25] s'appuie sur des hypothèses de symétrie du canal et des fonctions utilisées aux NV et aux NC pour calculer les messages. A partir de ces hypothèses de symétrie, on peut montrer que la probabilité d'erreur $P_e^{(\ell)}$ est indépendante du mot de code transmis. En conséquence, pour exprimer l'évolution de densité, on suppose en général que c'est le mot de code "tout à zéro" qui a été transmis, c'est à dire que $\mathbf{x} = \mathbf{0}$. Cela permet de simplifier grandement les expressions successives des distributions des messages.

Une deuxième hypothèse sur laquelle s'appuie l'évolution de densité est que le graphe de décodage à l'itération ℓ est un arbre, et ne contient donc aucun cycle. Pour ℓ fini, on peut effectivement montrer que la probabilité que le graphe de décodage soit un arbre tend vers 1 quand la longueur n du code tend vers l'infini. Sous cette hypothèse, les messages arrivant à un NV ou à un NC sont indépendants, ce qui simplifie également l'évaluation de la ditribution de probabilité de messages en sortie des noeuds.

Formulation générale de l'évolution de densité

Décrivons maintenant comment obtenir les formules d'évolution de densité à partir des hypothèses précédentes. Pour simplifier, nous considérons ici le cas d'un ensemble de codes réguliers de degrés (d_v, d_c) [25]. L'évolution de densité peut ensuite être étendue facilement au cas de codes irréguliers [26], ou modifiée pour considérer des protographes [36, Section 2.2.1]. Supposons dans un premier temps un décodeur LDPC discret, défini par une fonction Ψ_c pour les NC, et Ψ_v pour les NC. Avec cette notation, on suppose qu'un NC calcule un message de sortie $u_{d_c} = \Psi_v(\mathbf{t})$, où $\mathbf{t} = (t_1, t_2, \cdots, t_{d_c-1})$ représente les

$d_c - 1$ messages entrants. De même, on suppose qu'un NV calcule un message de sortie $t_{d_v} = \Psi_v(\mathbf{u})$, où $\mathbf{u} = (u_0, u_1, u_2, \cdots, u_{d_v-1})$. On suppose enfin que les messages u_{d_c} et t_v prennent leurs valeurs dans un ensemble discret $\mathcal{M} = [\![-q, +q]\!]$. Cette formalisation permet par exemple de représenter le décodeur de type offset MS quantifié présenté dans la Section 3.3.2.

L'objectif de l'évolution de densité est d'exprimer les distributions de probabilités $\mathbb{P}(u^{(\ell)})$ et $\mathbb{P}(t^{(\ell)})$ à chaque itération $\ell \in [\![1, L]\!]$. Pour un décodeur discret, en utilisant l'hypothèse d'indépendance des messages entrants, on montre que

$$\mathbb{P}(u^{(\ell)}) = \sum_{\mathbf{t}^{(\ell)} \in \mathcal{M}^{d_c-1} : \Psi_c(\mathbf{t}^{(\ell-1)}) = u^{(\ell)}} \prod_{c=1}^{d_c-1} \mathbb{P}(t_c^{(\ell-1)}) \tag{3.15}$$

$$\mathbb{P}(t^{(\ell)}) = \sum_{\mathbf{u}^{(\ell)} \in \mathcal{M}^{d_v} : \Psi_v(\mathbf{u}^{(\ell)}) = t^{(\ell)}} \mathbb{P}(u_0) \prod_{v=1}^{d_v-1} \mathbb{P}(u_v^{(\ell)}) \tag{3.16}$$

où $\mathbb{P}(u_0)$ est la distribution des messages initiaux calculée à partir de (3.12) et de l'hypothèse $\mathbf{x} = \mathbf{0}$. Les expressions récursives (3.15) et (3.16) peuvent se calculer de manière exacte. De plus, il existe quelques astuces pour réduire la complexité d'évaluation, voir par exemple ce qui est décrit dans [37].

On peut ensuite calculer la probabilité d'erreur $P_e^{(\ell)}$ des messages à chaque itération de la manière suivante :

$$P_e^{(\ell)} = \sum_{t^{(\ell)} < 0} \mathbb{P}(t^{(\ell)}) + \frac{1}{2}\mathbb{P}(0). \tag{3.17}$$

La condition d'erreur $t^{(\ell)} < 0$ vient du fait que l'on a considéré la transmission du mot de code $\mathbf{x} = 0$.

La probabilité d'erreur $P_e^{(\ell)}$ représente la fraction de messages incorrects dans le décodeur, c'est à dire menant à une mauvaise décision sur le bit correspondant du mot de code. Cette fraction de messages incorrects est déterminée vis-à-vis de l'ensemble des codes réguliers de mêmes paramètres (d_v, d_c). Le résultat de concentration fournit dans [25] indique que quand la longueur du code n tend vers l'infini, et pour tout $\delta > 0$, la probabilité que pour une instance particulière de code la probabilité d'erreur du code soit en dehors d'un intervalle $[P_e^{(\ell)} - \delta, P_e^{(\ell)} + \delta]$ converge vers 0. En d'autres termes, asymptotiquement, tous les codes de degrés (d_v, d_c) atteignent exactement la même performance de décodage. Mais ce résultat n'est pas nécessairement vérifié pour un n fixé.

Evolution de densité pour un décodeur avec des messages réels

La méthode d'évolution de densité décrite dans la partie précédente concerne les décodeurs échangeant des messages discrets. Pour les décodeurs utilisant des messages à valeurs réelles, comme le BP, il existe trois familles de solutions pour réaliser l'évolution de densité. La première consiste à réaliser l'évolution de densité en considérant que les messages sont quantifiés sur un grand nombre de bits [34]. Cela constitue une approche relativement peu complexe, au prix d'une petite perte en précision dans la prédiction de la probabilité d'erreur du décodeur. La deuxième approche consiste à réaliser des simulations de Monte Carlo en générant un grand nombre de messages à chaque itération, puis en leur appliquant les fonctions Ψ_v et Ψ_c du décodeur [38]. On estime ensuite les distributions de probabilité des messages en sortie des noeuds par des histogrammes. Cette approche plus complexe a également démontré son efficacité, et peut aussi être employée pour les cas où la quantification est plus difficile à définir, par exemple pour des décodeurs non-binaires. Enfin, une solution plus ancienne et un peu moins précise réside dans l'utilisation d'Exit Charts, qui suppose que les messages échangés dans le décodeur suivent une distribution Gaussienne [39, 40].

Seuil d'un ensemble de codes

La probabilité d'erreur $P_e^{(\ell)}$ est calculée pour des valeurs de d_v et d_c particulières, mais aussi en fonction des paramètres du canal. Par exemple, pour un canal BSC de paramètre p, on peut noter $P_e^{(\ell)}(p)$ pour plus de précision. Pour caractériser la performance d'un ensemble de codes de mêmes paramètres (d_v, d_c), on utilise en général la notion de "seuil" de l'ensemble de codes, défini comme le pire paramètre de canal pour lequel la probabilité d'erreur $P_e^{(\ell)}$ tend vers 0 quand n tend vers l'infini. Pour un BSC, le seuil est donc défini comme

$$\overline{p} = \max p \quad \text{tel que} \quad \lim_{n\to\infty} P_e^{(\ell)}(p) = 0$$

De même, pour un canal BiAWGN, le seuil est défini comme

$$\overline{\sigma^2} = \max \sigma^2 \quad \text{tel que} \quad \lim_{n\to\infty} P_e^{(\ell)}(\sigma^2) = 0.$$

Pour le canal BiAWGN, le seuil peut être exprimé de manière équivalente sur le rapport signal-à-bruit du canal.

3.4.2 Estimation des performances du code à longueur finie

La méthode d'évolution de densité décrite précédemment s'appuie sur des conditions asymptotiques. En pratique, il peut être utile de prédire la performance d'un décodeur LDPC pour une longueur n finie, ce que permet de faire la méthode proposée dans [41, 42]. Ainsi, pour un canal BSC de paramètre p, la probabilité d'erreur $P_{e,n}^{(\ell)}(p)$ pour une longueur n donnée peut s'exprimer comme [42]

$$P_{e,n}^{(\ell)}(p) = \int_0^{1/2} P_e^{(\ell)}(z)\phi_{\mathcal{N}}\left(z; p, \frac{p(1-p)}{n}\right) dz. \tag{3.18}$$

Dans cette expression, $P_e^{(\ell)}(z)$ représente la probabilité d'erreur (3.17) calculée à partir de l'évolution de densité asymptotique. De plus, $\phi_{\mathcal{N}}(z, m, \sigma^2)$ représente la densité de probabilité d'une variable aléatoire Gausienne de moyenne m et de variance σ^2. Cette expression est obtenue en considérant que l'estimation $\hat{p} = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)$ du paramètre p du canal est une variable aléatoire Gausienne de moyenne p et de variance $\frac{p(1-p)}{n}$ [43, Chapitre 8].

Il est aussi possible d'exprimer la probabilité d'erreur $P_{e,n}^{(\ell)}(\sigma^2)$ à longueur n pour un canal de type BiAWGN de variance σ^2 [41]. Dans ce cas, on applique la formule (3.18) avec $p = \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left(\frac{1}{\sqrt{2\sigma^2}}\right)$, où erf est la fonction d'erreur pour la distribution Gausienne. De plus, dans ce cas, on évalue la probabilité d'erreur asymptotique $P_e^{(\ell)}(z)$ pour une variance v^2 donnée par $v^2 = \frac{1}{2\left(\mathrm{erf}^{-1}(1-2z)\right)^2}$.

L'expression de $P_{e,n}^{(\ell)}(p)$ prend en compte la variabilité du canal pour une longueur n donnée. Bien que cette expression ne considère pas la présence de cycles dans le code, elle permet de prédire de manière précise la performance d'un ensemble de codes de paramètres (d_v, d_c) pour une longueur n assez grande. Dans les différents tests que nous avons effectué, nous avons observé que cette prédiction était tout à fait correcte pour des longueurs n au dessus de 5000 bits [44].

En conséquence, en fonction des usages, nous utiliserons soit l'évolution de densité asymptotique, soit la méthode de prédiction à longueur finie. Par exemple, pour optimiser la distribution des degrés ou le protographe pour un critère de performance uniquement, nous utilisons la version asymptotique, qui préserve l'ordre des distributions ou des protographes. En d'autres termes, si la distribution (λ_1, ρ_1) a une probabilité d'erreur asymptotique $P_e^{(\ell)}$ (3.17) plus faible que celle de la distribution (λ_2, ρ_2), on peut montrer que sa probabilité d'erreur $P_{e,n}^{(\ell)}$ calculée à partir de (3.18) sera aussi plus faible. En re-

vanche, quand il s'agit d'optimiser l'énergie d'un décodeur, nous verrons dans la suite que la méthode à longueur finie a plus de sens.

3.5 Construction de codes LDPC à longueur finie

L'évolution de densité permet de prédire la performance d'un décodeur donné, et d'optimiser la distribution des degrés ou le protographe du code. Il est ensuite nécessaire de construire la matrice de parité H du code pour une longueur n donnée. Cette étape a aussi une forte influence sur la performance finale du code, car les cycles courts présents dans la matrice peuvent fortement dégrader la performance de décodage.

Pour effectuer la construction d'un code LDPC pour une longueur n et un rendement R donné, j'emploie généralement les grandes étapes suivantes :

1. *Recherche d'une distribution des degrés* $(\lambda(x), \rho(x))$ *ou d'un protographe* B *avec le meilleur seuil possible pour le rendement* R, où le seuil est déterminé en utilisant l'évolution de densité asymptotique. L'optimisation du seuil est effectuée en utilisant un algorithme génétique appelé évolution différentielle [45], que l'on adapte pour la recherche d'une distribution des degrés [26] ou d'un protographe [36, Chapitre 4].

2. *Premier lifting de la distribution des degrés ou du protographe*, d'un facteur Z_1, en utilisant un algorithme de type Progressive Edge Growth (PEG) [46]. Cette étape va permettre d'obtenir une matrice de base de girth (longueur du plus petit cycle) la plus élevée possible. Pour une distribution des degrés $(\lambda(x), \rho(x))$, la matrice de base est de dimension $Z_1 \times \lceil RZ_1 \rceil$. Pour un protographe, la matrice de base est de dimension $(Z_1 m_c) \times (Z_1 n_v)$.

3. *Deuxième lifting Quasi-Cyclique*, d'un facteur Z_2, qui consiste à remplacer chacune des composantes non-nulles de la matrice de base par une matrice circulante de dimension $Z_2 \times Z_2$. A cette étape, on cherche également à augmenter la girth de la matrice de parité H finale. Pour cela, on utilise le résultat de [47, Théorème 2.1], qui donne une condition sur le choix des matrices circulantes qui permet d'éliminer les cycle inférieurs à une certaine longueur. Pour implémenter cette deuxième étape de lifting, on peut par exemple adapter l'algorithme présenté dans [48, Section IV.D] proposé à l'origine pour le choix des coefficients d'une matrice de parité d'un code LDPC non-binaire.

3.6 Conclusion

Dans ce chapitre, nous avons introduit les codes LDPC, ainsi que leur construction et leur décodage. Nous avons aussi présenté en détails la méthode de l'évolution de densité, qui permet de prédire la performance asymptotique ou à longueur finie d'un code. Dans la suite du document, nous allons considérer des applications particulières, comme les communications interactives ou l'implémentation d'algorithmes sur des architectures de calcul bruitées. Nous décrirons comment les codes LDPC peuvent être utilisés dans ces applications, et comment ils doivent être adaptés aux contraintes spécifiques de ces applications.

Chapitre 4

Codage de sources pour les communications interactives

4.1 Introduction

On considère un ensemble de J sources $\mathcal{V} = (X_{(1)}, X_{(2)}, \cdots, X_{(J)})$, présentant des dépendances statistiques les unes avec les autres. On suppose que ces N sources sont stockées sur un serveur, et que des utilisateurs veulent chacun accéder à un sous-ensemble $\mathcal{V}_k$ particulier de ces sources, où $\mathcal{V}_k \subseteq \mathcal{V}$ est le sous-ensemble souhaité par le k-ième utilisateur. Les sous-ensembles demandés par les utilisateurs sont inconnus *a priori*, c'est à dire au moment de la conception du système de compression. L'objectif est alors de concevoir un système de compression qui permettrait de minimiser à la fois le débit de stockage des sources sur le serveur, et le débit moyen de transmission des ensembles $\mathcal{V}_k$ entre le serveur et chacun des utilisateurs. Dans la suite, nous désignerons ce type de système par "codage de sources pour les communications interactives".

Dans ces conditions, on peut identifier deux stratégies opposées. La première stratégie consiste à coder l'ensemble des sources de manière conjointe, à un débit de stockage $H(X_{(1)}, X_{(2)}, \cdots, X_{(J)})$ correspondant à l'entropie jointe entre les sources [49, Chapitre 2]. Dans ce cas, le débit de transmission sera donné également par l'entropie jointe, ce qui est très peu efficace du point de vue de la transmission des données à l'utilisateur. A l'opposée, on peut choisir de coder chaque sous-ensemble possible de sources (soit 2^J sous-ensembles à stocker), ce qui est peu efficace du point de vue stockage mais optimal pour la transmission. Ces deux schémas ont déjà été implémentés dans des applications pratiques [9, 10], et illustrent le compromis qui existe entre le stockage et la transmission des données, que nous allons étudier dans ce chapitre.

On peut noter qu'une stratégie intermédiaire consisterait à coder conjointement l'ensemble des sources, et à effectuer un ré-encodage au niveau du serveur pour transmettre

uniquement les sources demandées par l'utilisateur. Mais nous avons choisi de ne pas considérer cette stratégie, qui est extrêmement coûteuse en en capacités de calcul. Nous supposerons donc que le serveur ne peut effectuer que des opérations peu complexes d'extraction d'information avant la transmission des sources demandées par l'utilisateur. Concrètement, cela signifie que si le serveur effectue une compression conjointe de manière à former une unique suite de bits représentant l'ensemble des sources, il pourra uniquement extraire une partie de cette suite de bits pour la transmettre à un utilisateur.

4.1.1 Examples d'applications

Nous présentons maintenant deux exemples d'applications du problème précédent.

Données collectées par des réseaux de capteurs

Le premier "toy exemple" auquel on peut penser est un problème de réseaux de capteurs, où les capteurs collectent des mesures météorologiques (température, humidité, etc.) sur une zone géographique donnée. Les mesures effectuées par les différents capteurs présentent des dépendances statistiques et sont stockées sous forme compressée sur un serveur. Des utilisateurs souhaitent accéder à un sous-ensemble de ces mesures : par exemple, sur une zone géographique plus concentrée (aux alentours de leur habitation), ou sur une fenêtre de temps donnée (sur une année en particulier). Lors de la compression, il est donc nécessaire de prendre en compte les corrélations qui existent entre les données à différentes échelles.

Images à 360°

Un deuxième exemple, qui correspond à des volumes de données plus importants, est le codage d'images à 360°, utilisées dans des applications de réalité virtuelle. Un exemple d'image à 360° projetée sur un plan est montré en Figure 4.1. En général, l'utilisateur ne souhaite pas avoir accès à l'ensemble de l'image en une seule fois, mais va effectuer une navigation dans l'image, accédant à une partie de l'image après l'autre, de manière fluide et continue. Là aussi, l'exploitation des corrélations qui existent dans les différentes parties de l'image devrait permettre d'effectuer une compression efficace des données. Mais les temps de traitement sont fondamentaux dans cette application, rendant impossible toute opération de ré-encodage.

FIGURE 4.1 – Projection sur un plan d'une image à 360°, provenant de la base de donnée décrite dans [50]

Dans ce deuxième exemple, qui a été bien étudié dans la littérature, la solution typique pour éviter de transmettre l'ensemble de l'image consiste à découper l'image en tuiles et à transmettre uniquement les tuiles d'intérêt pour l'utilisateur [51]. Il existe alors un compromis sur la taille des tuiles, entre l'efficacité de compression (dans le cas de grandes tuiles) et l'efficacité de transmission (dans le cas de petites tuiles). Mais cette solution ignore complètement les corrélations entre les tuiles.

4.1.2 Modélisation du problème

Pour simplifier le problème, nous avons supposé que l'accès aux sources s'effectue de manière séquentielle : même si l'utilisateur demande un sous-ensemble de sources en une fois, les sources lui seront transmises l'une après l'autre. Pour la transmission de la j-ème source de la requête, les sources déjà reçues peuvent-être exploitées côté utilisateur pour réduire la quantité d'informations à transmettre sur la source j.

Cette hypothèse supplémentaire nous a permis de traiter le problème en deux temps :

1. **Du point de vue d'une seule source** : On considère le stockage et la transmission d'une unique source X, pour un ensemble d'informations adjacentes possibles $\{Y_{(1)}, Y_{(2)}, \cdots, Y_{(K)}\}$ disponibles au décodeur. Les informations adjacentes possibles représentent une prédiction de la source X à transmettre, construite à partir des sources transmises précédemment. Le schéma de codage correspondant est décrit en Figure 4.2. Avec cette modélisation, qui peut-être rattachée à beaucoup de travaux

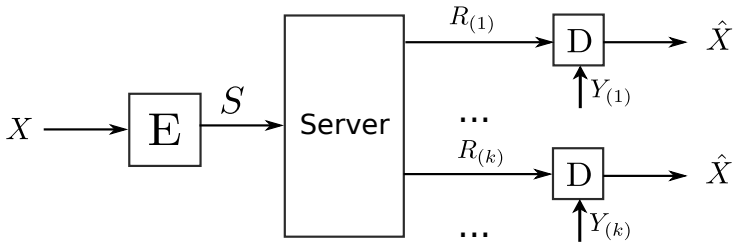

FIGURE 4.2 – Schéma de codage pour les communications interactives. La source X est compressée et stockée sur un serveur. Les utilisateurs veulent tous accéder à X, mais ils disposent d'informations adjacentes $Y_{(k)}$ différentes.

existants dans la littérature de la théorie de l'information et du codage, nous avons pu formellement définir le débit de stockage sur le serveur, et les débits de transmission du serveur aux utilisateurs. Nous avons ensuite effectué une analyse au sens de la théorie de l'information des performances atteignables par ce système. Puis nous avons proposé des schémas de codage efficaces s'appuyant sur des codes LDPC, et nous les avons appliqués dans des systèmes de codage d'images à 360°.

2. **Pour l'ensemble des sources** : Une fois la première étape bien traitée, nous avons abordé le problème de la transmission d'un sous-ensemble de sources. Pour cela, nous avons tout d'abord introduit un graphe de navigation qui représente l'ensemble des transitions possibles entre les J sources. Puis nous avons proposé des méthodes d'optimisation de ce graphe, de manière à adresser le compromis entre le débit de stockage et le débit moyen de transmission. Nous avons également observé que les schémas pratiques proposés dans le cas d'une seule source pouvaient être ré-utilisés dans le cas d'un ensemble de sources.

Dans la suite du chapitre, je présente les contributions associées à chacune de ces deux étapes.

4.2 Analyse de performances du point de vue d'une seule source

Dans un premier temps, nous avons souhaité caractériser la performance du système des communications interactives pour une seule source X avec K informations adjacentes possibles $(Y_{(1)}, \cdots, Y_{(K)})$. Dans cette partie, nous présentons d'abord le principe du codage de sources avec une seule information adjacente Y. Nous introduisons ensuite une

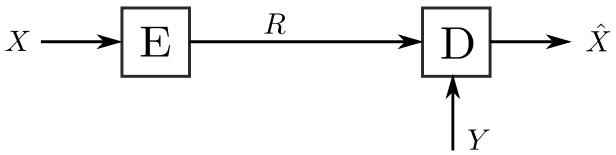

FIGURE 4.3 – Codage de sources avec information adjacente au décodeur

définition formelle de notre schéma de codage pour les communications interactives, et nous caractérisons ses performances à l'aide de la théorie de l'information.

4.2.1 Codage de sources avec information adjacente

Le codage de sources avec information adjacente, étudié par Slepian et Wolf dans le cas sans pertes [22], et par Wyner et Ziv dans le cas avec pertes [23], constitue un point de départ fondamental pour l'étude du problème des communications interactives du point de vue d'une seule source. Dans ce type de schéma de compression, représenté en Figure 4.3, il n'y a qu'une seule information adjacente Y possible. Dans le cas sans pertes, on peut montrer que le débit R de codage de la source X vérifie [22]

$$R \geq H(X|Y) \text{ bits/symbole}, \tag{4.1}$$

où $H(X|Y)$ est l'entropie conditionnelle de X sachant Y. Notons que dans le problème de compression classique sans information adjacente, le débit R atteignable est donné par $H(X) \geq H(X|Y)$. Le résultat énoncé dans (4.1) montre donc que l'information adjacente Y permet de réduire le débit de codage de X, alors même que Y n'est pas connue à l'encodeur. Dans le cas avec pertes, la fonction débit-distortion est fournie dans [23], et son expression est connue pour des sources Gausiennes non i.i.d. [52], des mélanges de Gaussiennes [53, 54], ou encore des modèles de Markov cachés [55].

Ce problème a ensuite connu de nombreuses déclinaisons. Par exemple, le cas où la distribution jointe $\mathbb{P}(X, Y)$ n'est pas bien connue a été étudié dans [56, 57, 58], tandis que le cas de sources non i.i.d. a été traité dans [59, Chapitre 7] et [60]. Une autre extension importante correspond au cas du codage de sources distribué, où plusieurs sources sont encodées séparément et transmises à un décodeur qui doit reconstruire l'ensemble des sources [61, 62].

4.2.2 Définition du schéma de codage

Nous considérons maintenant le cas de K d'utilisateurs, chacun avec une information adjacente $Y_{(k)}$ particulière, comme sur la Figure 4.2. Nous supposons de plus que toutes les distributions de probabilité jointes $P(X, Y_{(k)})$ sont parfaitement connues. De plus, nous allons introduire deux types de débits : un débit de stockage S de la source X sur le serveur, et des débits de transmission $R_{(k)}$ de la source X aux différents utilisateurs. Nous définissons maintenant de manière formelle le schéma de codage qui fait intervenir ces deux types de débits.

Definition 1 ([63]) *Un code pour l'ensemble de sources* $\{X, \{Y_{(k)}\}_{1\leq k\leq K}\}$ *est composé de :*

— *un* encodeur serveur h^{s} *qui assigne une séquence de* nS *bits à chaque vecteur* $\mathbf{x}^n \in \mathcal{X}^n$ *:*

$$h^{\mathrm{s}} : \mathcal{X}^n \to \{0,1\}^{nS} \tag{4.2a}$$

$$\mathbf{x}^n \mapsto (b_1, \ldots, b_{nS}). \tag{4.2b}$$

— *un ensemble de* K extracteurs $h^{\mathrm{ext}}_{(k)}$, $k \in [\![1, K]\!]$, *qui extraient une sous-séquence de bits à partir de la séquence de bits* $(b_1, \ldots, b_{nS})$ *:*

$$h^{\mathrm{ext}}_{(k)} : \{0,1\}^{nS} \to \{0,1\}^{nR_{(k)}} \tag{4.3a}$$

$$(b_1, \ldots, b_{nS}) \mapsto (b_j)_{j\in\mathcal{I}_{(k)}} \tag{4.3b}$$

où $\mathcal{I}_{(k)} \subseteq \{1, \cdots, nS\}$, *et* $|\mathcal{I}_{(j)}| = nR_{(k)}$.

— *un ensemble de* K décodeurs $g_{(k)}$, $k \in [\![1, K]\!]$, *qui construisent un estimée* $\hat{\mathbf{x}}^n_{(k)}$ *à partir de la séquence de bits reçue et du vecteur* $\mathbf{y}^n_{(k)}$ *:*

$$g_{(k)} : \{0,1\}^{nR_{(k)}} \times \mathcal{Y}^n_k \to \mathcal{X}^n \tag{4.4a}$$

$$(b_j)_{j\in\mathcal{I}_{(k)}}, \mathbf{y}^n_{(k)} \mapsto \hat{\mathbf{x}}^n_{(k)}. \tag{4.4b}$$

Dans cette définition, on voit ainsi apparaître le débit de stockage S de la source sur le serveur, et les débits de transmission $R_{(k)}$ du serveur à l'utilisateur k. Avec l'équation (4.3), on voit aussi que pour servir l'utilisateur k, l'encodeur peut uniquement extraire une partie de la suite de bits $(b_1, \cdots, b_{nS})$. Dans la suite, nous allons chercher à caractériser les débits

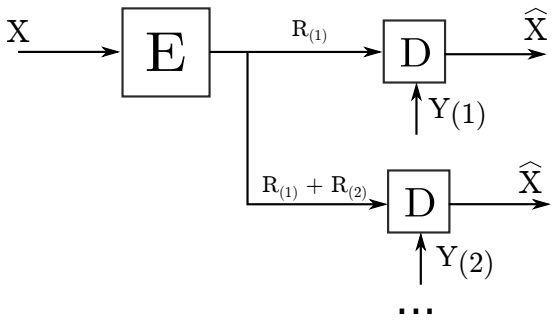

FIGURE 4.4 – Codage de sources avec successive refinement et information adjacente au décodeur

S et $R_{(k)}$ minimums atteignables par le système de compression pour les communications interactives, en fonction des distribution de probabilités jointes $\mathbb{P}(X, Y_{(k)})$.

4.2.3 Résultats existants

Certains résultats existants dans le domaine de la théorie de l'information peuvent être reliés au problème des communications interactives, bien qu'ils ne considèrent pas explicitement les deux types de débit S et $R_{(j)}$.

En particulier, le problème de "successive refinement" décrit en Figure 4.4 est celui qui est le plus proche de notre problème. Dans ce schéma, l'encodeur transmet au premier utilisateur un message $M_{(1)}$ à un débit $R_{(1)}$, puis au deuxième utilisateur un couple de messages $[M_{(1)}, M_{(2)}]$ à un débit $R_{(1)} + R_{(2)}$, et ainsi de suite. L'utilisateur k reçoit donc un vecteur de k messages $[M_{(1)}, M_{(2)}, \cdots, M_{(k)}]$ à un débit $R_{(1)} + R_{(2)} + \cdots + R_{(k)}$. Le problème de successive refinement adresse donc uniquement le problème des débits de transmission, en imposant comme nous une construction incrémentale des messages. Le problème de successive refinement sans pertes a été étudié dans [64]. Le cas avec pertes a été traité dans [65, 66, 67].

4.2.4 Région des débits atteignables pour le schéma de codage sans pertes

Dans cette partie, je présente les résultats de théorie de l'information que nous avons obtenu pour notre problème dans le cas sans pertes, en considérant la définition du schéma de codage introduite dans la Définition 1.

Sources i.i.d.

Pour des sources i.i.d., on peut utiliser des résultats du successive refinement [64] pour montrer que la région $(S, \{R_{(k)}\}_k)$ des débits atteignables est donnée par [63]

$$S \geq \max_{k\in[\![1,K]\!]} H(X|Y_{(k)}) \tag{4.5}$$

$$\forall k \in [\![1, K]\!], \quad R_{(k)} \geq H(X|Y_{(k)}). \tag{4.6}$$

Ce résultat montre que pour chaque information adjacente $Y_{(k)}$, on peut obtenir le débit optimal $H(X|Y_{(k)})$ que l'on aurait s'il n'y avait qu'un seul décodeur avec une information adjacente $Y_{(k)}$. Le débit de stockage donné par le maximum des $H(X|Y_{(k)})$ correspond au codage de la pire des sources.

Ce résultat est très intéressant dans la perspective de la construction de schémas de compressions pratiques pour les communications interactives. En effet, dans l'introduction du chapitre, nous avions évoqué deux stratégies utilisées actuellement dans les schémas pratiques existants. Dans le cas d'une seule source, la première stratégie, implémentée dans [9], consiste à stocker une seule version codée de X, à un débit $S = \max_{k\in[\![1,K]\!]} H(X|Y_{(k)})$, et à transmettre la source à un débit $R^{(k)} = \max_{k\in[\![1,K]\!]} H(X|Y_{(k)})$ également. La deuxième stratégie [10] consiste à stocker une représentation codée de X pour chaque information adjacente $Y_{(k)}$, ce qui correspond à $S = \sum_{k=1}^{K} H(X|Y_{(k)})$ et $R_{(k)} = H(X|Y_{(k)})$. Notre résultat énoncé dans cette section montre qu'une construction incrémentale du mot de code permet d'atteindre le meilleur débit de stockage possible $S = \max_{k\in[\![1,K]\!]} H(X|Y_{(k)})$ et les meilleurs débits de transmission possibles $R_{(k)} = H(X|Y_{(k)})$.

Sources générales

Dans [55], nous avons généralisé ce résultat à des sources non i.i.d., décrites pas des probabilités jointes $\mathbb{P}(\mathbf{X}^n, \mathbf{Y}^n_{(k)})$, pour tout $k \in [\![1, K]\!]$. Pour exprimer la région des débits atteignables dans ce cas, il est nécessaire de définir l'entropie conditionnelle spectrale $\bar{H}(\mathbf{X}|\mathbf{Y}_{(k)})$ de la manière suivante [68, Chapitre 7] :

$$\bar{H}(\mathbf{X}|\mathbf{Y}_{(k)}) = \text{p-}\limsup_{n\to\infty} \; -\frac{1}{n}\log \mathbb{P}(\mathbf{X}^n|\mathbf{Y}^n_{(k)}) \tag{4.7}$$

L'entropie conditionnelle spectrale est toujours définie, même si la distribution de probabilités conditionnelle $\mathbb{P}(\mathbf{X}^n|\mathbf{Y}^n_{(k)})$ ne converge pas. Elle permet donc de considérer des

modèles de sources très variés, mêmes non-stationnaires et non-ergodiques. A partir de cette définition, nous avons montré dans [63] que la région des débits atteignables pour des sources générales est donnée par

$$S \geq \max_{k \in [\![1,K]\!]} \bar{H}(\mathbf{X}|\mathbf{Y}_{(k)}),$$
$$R_{(k)} \geq \bar{H}(\mathbf{X}|\mathbf{Y}_{(k)}). \tag{4.8}$$

Dans la plupart des travaux sur les schémas de codage de type "successive refinement", on ne peut traiter que des sources physiquement dégradées, comme des canaux binaires symétriques avec des probabilités d'erreur p_k différentes. A l'inverse, le résultat que nous fournissons dans [63] ne nécessite pas cette condition. Pour réaliser le codage incrémental, il faut malgré tout ordonner les sources, de la "meilleure" à la "pire", et nous montrons que les sources peuvent être ordonnées en fonction de la valeur de l'entropie spectrale $\bar{H}(\mathbf{X}|\mathbf{Y}_{(k)})$. Cela permet de considérer des informations adjacentes de nature très différentes : par exemple, $Y_{(1)}$ pourrait être Gaussien, tandis que $Y_{(2)}$ serait Laplacien.

4.2.5 Région des débits atteignables pour le schéma de codage avec pertes

Nous avons ensuite souhaité traiter le cas avec pertes, plus réaliste dans de nombreuses configurations. Le cas général étant très complexe à traiter, nous nous sommes concentrés sur des sources Gaussiennes non i.i.d.. Pour cela, nous supposons que les vecteurs $\mathbf{X}^n, \mathbf{Y}^n_{(1)}, \cdots \mathbf{Y}^n_{(K)}$ sont tous centrés, de matrices de covariance Σ_X pour $\mathbf{X}^n$, et $\Sigma_{(k)}$ pour chaque $\mathbf{Y}^n_{(k)}$. On note $\Sigma_{X,Y_{(k)}} = [\mathbf{X}^n(\mathbf{Y}^n_{(k)})^T]$, et on définit les matrice de covariance conditionnelles de $\mathbf{X}^n$ sachant $\mathbf{Y}^n_{(k)}$:

$$\Sigma^n_{(X|k)} = \Sigma_X - \Sigma_{X,Y_{(k)}} \Sigma^{-1}_{(k)} \Sigma_{X,Y_{(k)}}. \tag{4.9}$$

Les valeurs propres de $\Sigma^n_{(X|k)}$ sont notées $\lambda_i^{(X|k)}$, $i \in [\![1, n]\!]$.

A partir de cette définition, nous avons obtenu dans [63] la région des débits attei-

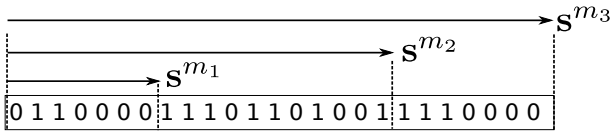

FIGURE 4.5 – Exemple de codage incrémental : la séquence de m_k bits $\mathbf{s}^{m_k}$ sera transmise si $Y_{(k)}$ est l'information adjacente disponible au décodeur.

gnables $\left(S, (R_{(k)}, D_{(k)})_{\in[\![1,K]\!]}\right)$ pour le cas avec pertes :

$$S \geq \max_{k\in[\![1,K]\!]} \lim_{n\to\infty} \sup \frac{1}{n}\sum_{i=1}^{n} \max\left(0, \frac{1}{2}\log_2 \frac{\lambda_i^{(X|k)}}{d_{(k),i}}\right) \tag{4.10}$$

$$\forall k \in [\![1,K]\!],\ R_{(k)} \geq \lim_{n\to\infty} \sup \frac{1}{n}\sum_{i=1}^{n} \max\left(0, \frac{1}{2}\log_2 \frac{\lambda_i^{(X|k)}}{d_{(k),i}}\right) \tag{4.11}$$

$$\forall k \in [\![1,K]\!],\ D_{(k)} \geq \lim_{n\to\infty} \sup \frac{1}{n}\sum_{i=1}^{n} \min\left(\lambda_i^{(X|k)}, d_{(k),i}\right), \tag{4.12}$$

où $d_{(k),i} = \frac{\lambda_i^{(X|k)}\delta}{\lambda_i^{(X|k)}+\delta}$. Comme dans le cas sans pertes, on voit que le débit de stockage S est donné par la pire des fonctions débit-distortion, tandis que le débit de transmission $R_{(k)}$ correspond à la fonction débit-distortion pour la source X sachant que $Y_{(k)}$ est disponible au décodeur.

4.3 Schéma de codage pratique

On souhaite maintenant construire un schéma de codage de sources dont la performance s'approche le plus possible des bornes théoriques précédentes. Dans cette partie, nous considérons uniquement des sources binaires i.i.d., et un codage sans pertes. Nous verrons dans la suite un exemple d'application au cas plus complexe des images à 360°.

Des schémas de codage sans perte efficaces ont été proposés pour le problème de codage de sources avec une seule information adjacente [69, 70, 71]. Ils utilisent tous des codes correcteurs d'erreurs introduits à l'origine dans le domaine du codage canal : des codes Turbo dans [69, 70] et des codes LDPC dans [71, 72]. Dans ces travaux, les structures d'encodage et de décodage des codes canal sont modifiées pour être adaptée au problème du codage sources. De plus, pour le problème des communications interactives, les preuves de théorie de l'information nous indiquent qu'il faut réaliser une construction incrémentale des différents mots de code, de manière à pouvoir atteindre l'entropie conditionnelle pour

chacune des informations adjacentes possibles [63]. Un exemple de code incrémental est représenté en Figure 4.5, sur laquelle on voit que les séquences binaires associées aux informations adjacentes successives $Y_{(1)}$ à $Y_{(3)}$ sont imbriquées.

Dans cette partie, nous décrivons tout d'abord l'utilisation de codes LDPC pour le problème de codage de sources avec une seule information adjacente. Nous montrons ensuite comment construire des codes LDPC incrémentaux pour les communications interactives.

4.3.1 Codes LDPC pour le codage de sources

Comme décrit dans [71, 72], les codes LDPC peuvent être utilisés pour construire des schémas de codage de sources avec une seule information adjacente Y, grâce à une simple adaptation que nous présentons maintenant. On considère un code LDPC décrit par sa matrice de parité binaire H de dimension $m \times n$, voir Chapitre 3. On souhaite coder un vecteur de source $\mathbf{x}^n$ binaire de longueur n. Pour cela, on calcule le syndrôme

$$\mathbf{s}^m = H\mathbf{x}^n, \tag{4.13}$$

qui constituera le mot de code transmis au décodeur. Il s'agit bien d'une compression, car $m < n$. On définit le débit de codage source, ou taux de compression, par $R_s = m/n$.

On peut ensuite utiliser n'importe quel décodeur LDPC (BP, MS, etc.), et l'adapter pour estimer $\mathbf{x}^n$ à partir du mot de code $\mathbf{s}^m$ et du vecteur d'information adjacente $\mathbf{y}^n$. Par exemple, pour le décodeur BP, en reprenant les notations de la Section 3.3.1, les équations de mise à jour des messages deviennent [71]

$$u_v^{(0)} = \log \frac{P(x_v = 0|y_v)}{P(x_v = 1|y_v)} = \log \frac{P(y_v|x_v = 0)P(x_v = 0)}{P(y_v|x_v = 1)P(x_v = 1)} \tag{4.14}$$

$$u_{c\to v}^{(\ell)} = (-1)^{s_c} \frac{1 + \prod\limits_{v' \in \mathcal{V}_c \setminus v} \tanh\left(t_{v'\to c}^{(\ell-1)}\right)}{1 - \prod\limits_{v' \in \mathcal{V}_c \setminus v} \tanh\left(t_{v'\to c}^{(\ell-1)}\right)} \tag{4.15}$$

$$t_{v\to c}^{(\ell)} = u_v^{(0)} + \sum_{c' \in \mathcal{C}_v \setminus c} u_{c'\to v}^{(\ell)} \tag{4.16}$$

Dans le décodeur "source", les messages initiaux $u_v^{(0)}$ sont calculés à partir des probabilités $P(x_v|y_v)$ au lieu de $P(y_v|x_v)$, car la source X n'est pas nécessairement symétrique dans le sens où $P(x_v = 0) \neq P(x_v = 1)$ en général. De plus, dans les messages des NC vers les NV, on voit apparaître un terme supplémentaire : $(-1)^{s_c}$, où s_c est la composante

du syndrôme $\mathbf{s}^m$ à la position c. Ce terme permet de prendre en compte la condition $\mathbf{s}^m = H\mathbf{x}^n$ (au lieu de $H\mathbf{x}^n = 0$ en codage de canal). Les messages des NV vers les NC restent, eux, inchangés.

Dans le cas du codage de sources, on peut ré-utiliser la majorité des outils de conception de codes développés pour le codage de canal. Si la source X est symétrique, l'évolution de densité mènera exactement aux mêmes expressions qu'en codage de canal, car l'hypothèse du mot de code tout à zéro sera alors toujours valable. Si la source X est asymétrique, on peut faire une évolution de densité en définissant cette fois un canal symétrique équivalent, voir [73] pour le cas binaire et [74] pour le cas non-binaire. De plus, une fois que la distribution des degrés ou le protographe est fixé, la méthode de construction de codes à longueur finie est identique, car les critères liés à la topologie du graphe (limiter le nombre de cycles courts, etc.) sont toujours pertinents.

4.3.2 Codes LDPC compatibles en rendement

Dans notre cas avec plusieurs informations adjcentes, on peut développer une construction incrémentale en utilisant des codes LDPC compatibles en rendement [28, 75]. Si pour le décodage et la construction de codes LDPC non-incrémentaux, il y a peu de différences entre la version "source" et la version "canal", les choses ne sont pas aussi simples lorsque l'on souhaite construire des codes compatibles en rendement. Pour illustrer cela, nous présentons ici un bref état de l'art de la construction de codes compatibles en rendement, d'abord en codage de canal puis en codage de sources.

Codes LDPC compatibles en rendement pour le codage canal

Lorsque l'on veut construire des codes LDPC compatibles en rendement pour le codage de canal, il existe deux grandes familles de solutions. La première consiste à poinçonner le code [28], c'est à dire à éliminer certains bits du mot de code $\mathbf{x}^n$ transmis sur le canal de communications. Si on voulait directement appliquer cette solution au codage de sources, c'est donc le syndrôme $\mathbf{s}^m$ qu'il faudrait poinçonner. Le problème est que le poinçonnage du syndrôme détruit assez rapidement la structure du code, et donne des performances de décodage très dégradées [76].

La deuxième grande famille de solutions consiste à étendre la matrice de parité H [75, 77], c'est à dire à ajouter progressivement des lignes et des colonnes suplémentaires à cette matrice. Malheureusement, il n'est pas possible d'appliquer directement cette solution en

codage de sources. En effet, si on peut étendre la dimension m du syndrôme, il n'est pas possible d'en faire de même pour la dimension n du vecteur de sources, puisque celui-ci est fixé. C'est pourquoi des solutions spécifiques au codage de source ont été proposées.

Codes LDPC compatibles en rendement pour le codage source

En codage de sources, il existe aussi deux grandes familles de méthodes pour construire des codes compatibles en rendement, en fonction du débit source R_s de la matrice H de départ. Si l'on part d'un code de faible débit R_s, on va utiliser une solution dite "rateless" [78, 79], qui consiste à ajouter au mot de code $\mathbf{s}^m$ des bits de la source $\mathbf{x}^n$. Cependant, cette méthode dépend fortement de la performance du code pour la matrice de parité initiale H. Or il est souvent difficile de construire des matrices H performantes pour des débits sources faibles [80].

A l'inverse, si l'ont part d'un débit source R_s élevé, on peut utiliser la méthode LDPCA (Low Density Parity Check Accumulate) [28]. Cette méthode consiste à calculer un syndrôme accumulé $\mathbf{a}^m$ de la manière suivante :

$$\begin{aligned} a_1 &= c_1 \\ a_i &= a_{i-1} \oplus s_i, \quad \forall i \in [\![1, m]\!]. \end{aligned} \tag{4.17}$$

Ensuite, pour diminuer le débit source, on transmet uniquement un sous-ensemble des bits de $\mathbf{a}^m$. Par exemple, si on souhaite obtenir un débit $R_s/2$, on transmet les bits pairs $a_2, a_4, \cdots$. Ensuite, le décodeur commencera par calculer $a_i - a_{i-2} = s_i \oplus s_{i-1}$, puis applique un décodeur LDPC standard comme le BP, pour reconstruire la source $\mathbf{x}^n$ à partir du graphe de Tanner modifié et correspondant aux équations de parités $s_i \oplus s_{i-1}$.

On peut voir la méthode LDPCA comme un poinçonnage du syndrôme accumulé $\mathbf{a}^m$, et [28] montre que les performances sont bien meilleures que lorsque l'on poinçonne directement $\mathbf{s}^m$. Le problème de cette méthode est que la structure proposée dans [28], et rappelée dans l'équation (4.17), est fixe, et ne permet donc pas d'optimiser la structure du code (distribution des degrés, présence de cycles courts, etc.) aux différents débits. Cela induit une forte dépendance de la méthode au code original. Une solution alternative a été proposée dans [81], qui consiste à choisir un entrelaceur différent de (4.17), de manière à optimiser la distribution des degrés des codes utilisés aux débits inférieurs à R_s. Mais la méthode d'optimisation s'appuie sur une évolution de densité sous des hypothèses asymptotiques. Elle ne permet donc pas de prendre en compte les critères habituellement

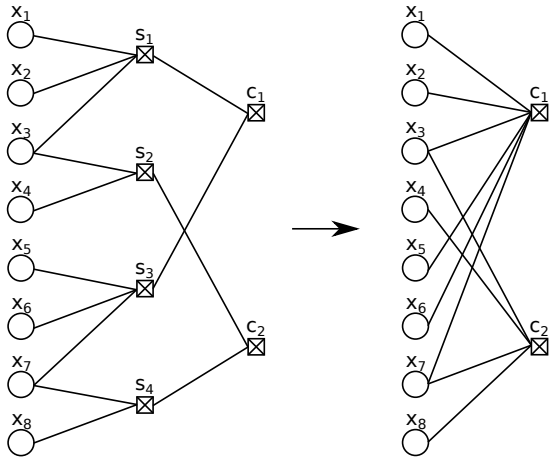

FIGURE 4.6 – Exemple de graphes de Tanner. Le graphe de gauche représente la matrice H_1 entre les VN x_i et les CN s_j, suivi de la matrice $H_{1\to 2}$ entre les CN s_j et les CN c_k. Le graphe de droite représente la matrice résultante H_2 entre les VN x_i et les CN c_k.

considérés à longueur finie, comme la présence de cycles courts.

4.3.3 Construction incrémentale proposée

Nous proposons maintenant une nouvelle méthode de construction de codes sources LDPC compatibles en rendement, permettant d'optimiser non seulement la distribution des degrés ou le protographe des codes aux différents débits, mais aussi de maîtriser la topologie des graphes de Tanner à chaque débit. Cette méthode a été décrite en détails dans [82] pour des codes irréguliers binaires, dans [83] pour des codes irréguliers non-binaires, et dans [84] pour des constructions binaires à base de protographes. C'est cette dernière approche que je présente maintenant. Dans la suite, nous considérons tout d'abord le passage d'un débit R_1 à un débit $R_2 < R_1$. La généralisation à d'autres débits sera ensuite brièvement évoquée.

Construction du code

Dans cette partie, on note H_1 la matrice de parité du code initial, ou code mère, de dimension $n \times m_1$ et de débit R_1. Le graphe de Tanner associé à H_1 est noté $\mathcal{T}_1$ et connecte les NV $\mathcal{X} = \{x_1, \cdots x_n\}$ aux NC $\mathcal{S} = \{s_1, \cdots, s_{m_1}\}$. A partir de cette matrice mère, on souhaite construire une matrice fille, de dimension $n \times m_2$ et de débit $R_2 < R_1$. Le graphe de Tanner associé à H_2 est noté $\mathcal{T}_2$ et connecte les NV $\mathcal{X}$ aux NC $\mathcal{C} = \{c_1, \cdots, c_{m_2}\}$.

Pour lier les matrices H_1 et H_2, on suppose qu'il existe une matrice intermédiaire $H_{1\to 2}$ de dimension $m_2 \times m_1$ telle que

$$H_2 = H_{1\to 2} H_1. \tag{4.18}$$

Le graphe de Tanner $\mathcal{T}_{1\to 2}$ de $H_{1\to 2}$ connecte donc les NC $\mathcal{S}$ de $\mathcal{T}_1$ aux NC $\mathcal{C}$ de $\mathcal{T}_2$, voir un exemple en Figure 4.6. Il faut donc construire la matrice intermédiaire $H_{1\to 2}$ de manière à obtenir de bonnes performances de décodage pour H_2, mais aussi de manière à obtenir une adaptation en débit entre H_1 et H_2, comme on le décrit maintenant.

Condition pour l'adaptation en débit

Pour réaliser l'adaptation en débit, nous proposons d'utiliser les règles suivantes. Pour transmettre la source au débit R_2 le plus faible, on transmet complètement le mot de code $\mathbf{c}^{m_2}$, obtenu après application de la matrice H_2, qui est utilisée aussi pour réaliser le décodage. Pour transmettre la source au débit R_1 plus élevé, on transmet à la fois le mot de code $\mathbf{c}^{m_2}$, et $m_2 - m_1$ bits du mot de code $\mathbf{s}^{m_1}$. Formellement, on dit que l'on transmet un sous-ensemble $\mathcal{S}' \subseteq \mathcal{S}$, en plus de $\mathcal{C}$. Dans ce cas, pour pouvoir utiliser la matrice H_1 pour réaliser le décodage, il faut pouvoir reconstruire complètement l'ensemble des bits contenus dans $\mathcal{S}$ à partir de $\mathcal{S}'$ et $\mathcal{C}$.

Les ensembles $\mathcal{S}'$ et $\mathcal{C}$ définissent un système de m_1 équations à m_1 inconnues $\mathcal{C}$. On dit alors que le code $(H_1, H_{1\to 2}, \mathcal{S}')$ est adaptable en débit si le système précédent a une unique solution. Dans [84], on montre que si la matrice $H_{1\to 2}$ est de rang plein, alors il est toujours possible de trouver un ensemble $\mathcal{S}'$ tel que le code $(H_1, H_{1\to 2}, \mathcal{S}')$ soit adaptable en débit. Au débit de codage R_1, on peut donc reconstruire complètement $\mathcal{S}$, de sorte que la performance de décodage ne dépende ensuite que de H_1. Par la suite, la difficulté principale reste de construire la matrice H_2 à partir de H_1, c'est à dire de choisir la matrice $H_{1\to 2}$ de manière à assurer une bonne performance de décodage pour H_2 également.

4.3.4 Construction du code

On souhaite maintenant construire les matrices H_1 et $H_{1\to 2}$ de manière à obtenir de bonnes performances de décodages à la fois pour les débits R_1 et R_2.

La matrice H_1 est construite en premier, à partir de la méthode décrite dans la Section 3.5. On note B_1 le protographe de la matrice H_1, et B_1 est obtenu par optimisation du seuil calculé par évolution de densité. Pour construire la matrice H_2 donnée par (4.18), nous avons montré dans [84], que sous certaines hypothèses, la relation suivante était

valide :

$$B_2 = B_{1\to 2} B_1, \tag{4.19}$$

où B_2 est le protographe de la matrice H_2, et $B_{1\to 2}$ est le protographe de la matrice intermédiaire $H_{1\to 2}$. On peut constater que la relation (4.19) est la version "protographe" de la relation (4.18) sur les matrices. L'intérêt de cette relation, est que l'on va donc pouvoir choisir le protographe intermédiaire $B_{1\to 2}$ de manière à optimiser le seuil du protographe B_2. Ensuite, nous pourrons construire la matrice $H_{1\to 2}$ correspondant au protographe $B_{1\to 2}$, en adressant d'autres critères, de cycles courts notamment.

Pour construire la matrice H_2, on procède donc en deux étapes :

1. Après avoir introduit certaines contraintes sur le choix du protographe $B_{1\to 2}$ (valeur maximum des composantes, nombre maximal de valeurs non-nulles sur chaque colonne, etc.), on liste l'ensemble des possibilités pour ce protographe, et on calcule le seuil du protographe B_2 résultat pour chaque protographe intermédiaire possible. On garde alors le protographe intermédiaire $B_{1\to 2}$ qui maximise le seuil de B_2.
2. Le protographe intermédiaire $B_{1\to 2}$ indique le type de lignes à combiner dans la matrice H_1, pour obtenir une matrice H_2 qui correspond au protographe B_2. On met en oeuvre un algorithme de type PEG pour choisir les lignes à combiner en respectant les contraintes de $B_{1\to 2}$, de manière à minimiser le nombre de cycles courts dans la matrice H_2 finale.

4.3.5 Généralisation à plusieurs débits

Pour pouvoir utiliser la construction proposée dans une application pratique, on souhaite pouvoir utiliser n'importe quel débit inférieur à R_1. Pour cela, nous avons proposé la méthode suivante, qui s'appuie sur une notion de débits encres ("anchor rates") [84].

Pour expliquer cette construction, prenons l'exemple d'un code mère de débit $R_1 = 1/2$, avec un protographe B_1 de dimension 4×8. Si on veut utiliser la relation (4.19), on peut d'abord combiner deux lignes de B_1 pour obtenir un protographe B_2 de dimension 3×8, puis obtenir successivement des protographes B_3 et B_4, de dimensions respectives 2×8 et 1×8. Nous optimisons donc les protographes B_2, B_3, B_4, comme décrit dans l'étape 1 de la Section 4.3.4. Nous appelons les débits correspondants $R_2 = 3/8$, $R_3 = 1/4$, $R_4 = 1/8$, les débits encres, car ils correspondent à des protographes optimisés.

Une fois que ces protographes sont fixés, on peut obtenir n'importe quel débit intermédiaire en combinant deux à deux les lignes indiquées par les protographes correspon-

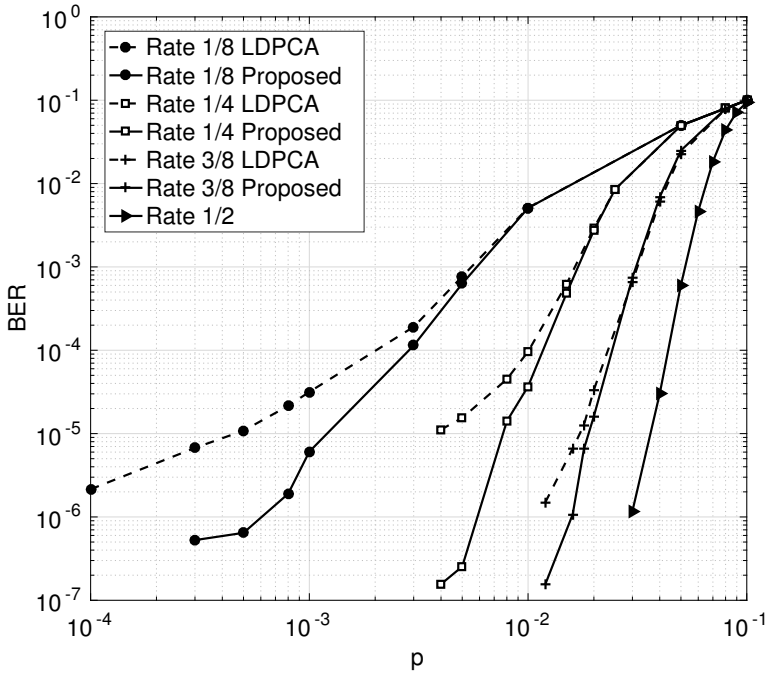

FIGURE 4.7 – Taux d'erreur binaire (BER) en fonction de p, pour différents débits de codage, pour un code mère de dimension 256×512. Comparaison entre la solution proposée et la méthode LDPCA.

dants $S_{1\to 2}$, $S_{2\to 3}$, $S_{3\to 4}$, en utilisant l'algorithme introduit à l'étape 2 la Section 4.3.4. La description du code adaptable en débit contient donc la matrice originale H_1, puis les combinaisons de lignes à effectuer pour obtenir l'ensemble des débits intermédiaires. Cela correspond à une granularité de $1/2n$ en débit. Par exemple, pour le débit $R = \frac{1}{2} - \frac{1}{2n}$, la matrice H contiendra $m-2$ lignes de H_1, et 1 ligne contenant la somme XOR des 2 lignes restantes de H_1.

4.3.6 Résultats de simulations

Nous présentons maintenant quelques résultats de simulations pour illustrer l'intérêt de la méthode proposée. On suppose que les sources X et Y sont i.i.d. et binaires, et que X suit une loi de Bernoulli de paramètre $1/2$. On suppose de plus que la source Y est obtenue par passage de X à travers un canal BSC de paramètre p. On considère un code

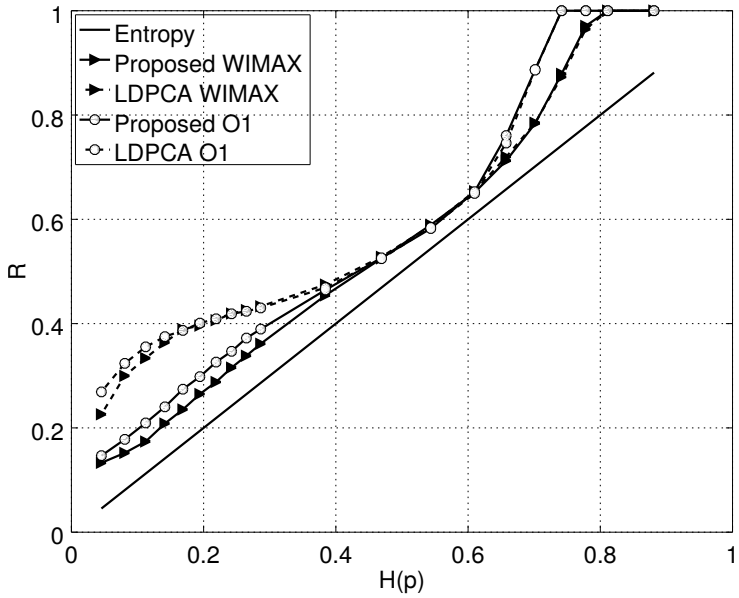

FIGURE 4.8 – Débits moyens en fonction de l'entropie $H(p)$, pour un code mère de dimension 256×512. Comparaison entre la solution proposée et les méthodes LDPCA/Rateless.

mère de matrice de parité H_1 de dimension 256×512, et de protographe

$$B = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{4.20}$$

En appliquant la méthode de construction incrémentale décrite dans la partie 4.3.4, nous avons construit les codes filles correspondants aux débits 3/8, 1/4, et 1/8. La Figure 4.7 montre la performance en termes de taux d'erreur binaire (BER) en fonction de p, pour les codes aux différents débits. Notre méthode de construction est comparée à la méthode LPDCA de l'état de l'art [76]. Pour tous les débits, on observe un gain significatif en terme de BER de notre méthode comparée au LDPCA.

On considère ensuite un deuxième protocole expérimental, pour le même modèle de source et pour 2 codes mères différents : un code Wimax de longueur $n = 192$, et le code précédent construit à partir de B. Pour chaque valeur de p considérée, on génère aléatoirement un grand nombre de vecteurs de sources $(\mathbf{x}^n, \mathbf{y}^n)$. Pour chaque paire, on utilise la construction incrémentale précédente et on choisit le débit R minimum qui permet d'assurer que le vecteur $\mathbf{x}^n$ est reconstruit sans erreur au décodeur. Pour les

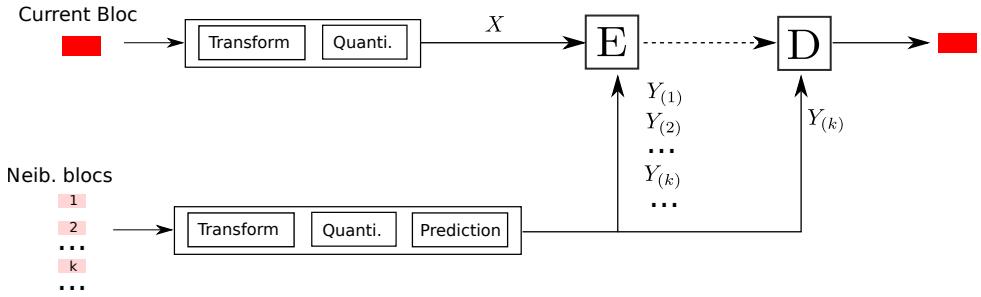

FIGURE 4.9 – Schéma de codage d'images à 360°

débits supérieurs à 1/2, on utilise la méthode rateless [78]. La Figure 4.8 représente les débits moyens calculés sur un grand nombre de vecteurs, en fonction de $H(p)$. On compare donc la performance de notre méthode au débit optimal $H(p)$, et à la méthode LDPCA. On observe non seulement un gain comparé au LDPCA, mais aussi un écart constant, au moins jusqu'à $R < 0.6$ entre la courbe de notre méthode et celle de l'entropie, ce qui montre que sa bonne performance et sa robustesse pour une large gamme de débits.

4.4 Application au codage d'images à 360 degrés

Les résultats de simulation ayant démontré la bonne performance des constructions incrémentales à base de codes LDPC, nous avons souhaité les appliquer à la compression d'images à 360°. Dans la Section 4.1.1, nous avons vu que les solutions existantes découpent les images en blocs compressés séparément [51], mais n'exploitent pas les corrélations entre les blocs. Nous avons donc souhaité utiliser notre schéma de codage décrit précédemment pour construire des représentations compressées incrémentales de chaque bloc. En fonction des blocs déjà reçus par l'utilisateur, un mot de code d'une certaine longueur sera extrait de la représentation compressée pour permettre à l'utilisateur de décoder le bloc courant.

4.4.1 Données à disposition

Pour cela, nous sommes partis d'images à 360° projetées en deux dimensions, selon une méthode proposée par nos collaborateurs à INRIA Rennes [85]. Dans cette méthode, après projection, les images sont découpées en blocs de dimension 32×32, et on applique des opérations de transformation et de quantification à chacun des blocs. Suite à ces opérations, chaque bloc peut être vu comme une réalisation de la source X. Quand un

ou plusieurs blocs voisins de X sont déjà disponibles au décodeur, on peut construire une prédiction $Y_{(k)}$ de X à partir des blocs disponibles. Au moment de l'encodage, on peut donc considérer qu'il y a K prédictions possibles $Y_{(1)}, \cdots, Y_{(K)}$ pour la source X. Ce schéma est représenté en Figure 4.9.

L'équipe d'INRIA Rennes nous as fourni des séquences de symboles de longueur $n = 1024$, correspondant à des réalisations de X, et des séquences d'informations adjacentes possibles $Y_{(k)}$, avec $K = 8$ ou $K = 12$. On note qu'ici, les symboles générés par X et $Y_{(k)}$ sont discrets mais pas binaires, car ils s'agit directement des valeurs obtenues après quantification. Nous avons cherché à appliquer notre méthode de construction de codes LDPC adaptables en débit pour réaliser le codage entropique des séquences de manière incrémentale. Je décris ici les étapes successives pour arriver à ce résultat.

4.4.2 Choix du modèle de corrélation

La méthode de construction de codes décrite dans la Section 4.3.3 est générique et peut s'appliquer à n'importe quel modèle de corrélation $P(X, Y_{(k)})$ dans lequel à la fois la source X et l'information adjacente sont i.i.d. Ce modèle doit cependant être connu, que ce soit pour initialiser le décodeur (voir Section 4.3.1), ou pour optimiser la construction du code incrémental. Ici, nous supposons que la transformation appliquée à la source X permet d'éliminer l'ensemble des corrélations présentes dans le bloc, et de même pour l'information adjacente $Y_{(k)}$ [85]. Cela justifie l'hypothèse i.i.d. pour X comme pour les $Y^{(k)}$. Ensuite, nous avons considéré un modèle additif, dans lequel on suppose qu'il existe une variable aléatoire $Z_{(k)}$, indépendante de X, telle que $Y_{(k)} = X + Z_{(k)}$. Ce modèle additif est courant dans la littérature [86, 87, 88].

Il reste donc à déterminer un modèle statistique pour la variable aléatoire $Z_{(k)}$. Dans la littérature, c'est souvent un modèle Laplacien qui est employé [86, 89, 90]. Mais s'il existe une corrélation très forte entre X et $Y_{(k)}$, comme c'est parfois le cas dans les données que nous observons, le modèle Laplacien ne représente pas bien les petites valeurs de $Z_{(k)}$. Dans [91], nous avons proposé d'utiliser à la place un modèle q-aire symétrique, comme alternative au modèle Laplacien souvent employé dans la littérature [86, 89]. Ce modèle

est défini de la manière suivante :

$$P^{(k)}(z) = \begin{cases} q_k & \text{if } z = 0 \\ \frac{1-q_k}{Z_{k,\max}-Z_{k,\min}} & \text{if } z \neq 0 \text{ and } Z_{k,\min} \leq z \leq Z_{k,\max} \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

où $q_k \in [0,1]$, et $Z_{k,\min}$ et $Z_{k,\max}$ sont les valeurs minimum et maximum possibles pour $Z_{(k)}$. Ces trois paramètres sont estimés à l'encodeur, puis quantifiés sur un certain nombre de bits et transmis au décodeur avec le mot de code.

4.4.3 Schéma de codage incrémental

Nous souhaitons maintenant proposer un schéma incrémental pour le codage d'un vecteur de source $\mathbf{x}^n = [x_1, \cdots, x_n]$ sachant qu'un vecteur d'information adjacente $\mathbf{y}^n_{(k)} = [y_{(k),1}, \cdots, y_{(k),n}]$ est disponible au décodeur. Bien que les symboles $x_i, y_{(k),i}$ correspondent à des symboles discrets, nous avons choisi de construire le schéma de codage à partir de codes LDPC binaires, car ils sont plus simples à construire et ont une complexité de décodage significativement plus faible que les codes non-binaires [92]. Une première étape consiste donc à transformer les vecteurs de sources $\mathbf{x}^n$ en plan de bits, qui seront chacun codés avec un code LDPC binaire. Dans [91], nous avons montré que cette transformation binaire n'introduit aucune perte en débit de transmission, mais cause une petite perte en débit de stockage.

Concrètement, le vecteur $\mathbf{x}^n$ est transformé en $B+1$ plans de bits $\mathbf{q}^n_{(b)}$, $b \in [\![0, B]\!]$. Le plan de bits $b = B$ correspond aux signes, le plan de bits $b = B-1$ correspond aux bits de poids forts, et le plan de bits $b = 0$ correspond aux bits de poids faibles. Ensuite, le plan de bits $\mathbf{q}^n_{(b)}$ est codé à l'aide d'une matrice de parité binaire H, ce qui permet d'obtenir un mot de code $\mathbf{s}^m_{(b)}$ à partir de la relation suivante :

$$\mathbf{s}^m_{(b)} = H\mathbf{q}^n_{(b)}. \tag{4.22}$$

En réalité, on utilise la construction adaptable en débit décrite dans la Section 4.3.3 pour pouvoir extraire un mot de code $\mathbf{s}_{(b,k)}$ pour chaque vecteur d'information adjacente $\mathbf{y}^n_{(k)}$ possible. La longueur $m_{b,k}$ du mot de code $\mathbf{s}_{(b,k)}$ est choisie de manière à pouvoir décoder le plan de bits $\mathbf{q}^n_{(b)}$ sans erreurs à partir de $\mathbf{y}^n_{(k)}$.

Pour un vecteur d'information adjacente $\mathbf{y}_{(k)}$, le décodeur va donc recevoir B plans

de bits $\mathbf{s}_{(0,k)}$ à $\mathbf{s}_{(B,k)}$. Les plans de bits sont décodés les uns après les autres à l'aide du décodeur BP décrit dans la Section 4.3.1, en commençant par le plan de bits $\mathbf{q}^n_{(B)}$. Lors du décodage de $\mathbf{q}^n_{(b)}$, on souhaite utiliser l'information issue du décodage des plans de bits précédents $\widehat{\mathbf{q}}^n_{(b+1)} \cdots \widehat{\mathbf{q}}^n_{(B)}$. Pour cela, on initialise le décodeur BP avec les probabilités suivantes :

$$P\left(Q_{(b),i} = 0 \mid Y_i^{(k)} = y, \hat{q}_{(b+1),i}, \ldots, \hat{q}_{(B),i}\right) \tag{4.23}$$

$$P\left(Q_{(b),i} = 1 \mid Y_i^{(k)} = y, \hat{q}_{(b+1),i}, \ldots, \hat{q}_{(B),i}\right) \tag{4.24}$$

dont les expressions dépendent du modèle q-aire symétrique et sont données dans [91].

4.4.4 Estimation du débit

Nous exprimons maintenant les débits de stockage et de transmission que l'on obtient avec la méthode de construction précédente. On rappelle que pour le plan de bits $\mathbf{q}^n_{(b)}$, lorsque le vecteur d'information adjacente $\mathbf{y}^n_{(k)}$ est présent au décodeur, on transmet un mot de code de longueur $m_{b,k}$. Le débit de transmission $R_{(b,k)}$ pour la paire $(\mathbf{q}^n_{(b)}, \mathbf{y}^n_{(k)})$ et le débit de stockage $S_{(b)}$ pour $\mathbf{q}^n_{(b)}$ sont donnés par :

$$R_{(b,k)} = \frac{1}{n} \sum_{m=0}^{B} m_{b,k}, \tag{4.25}$$

$$S_{(b)} = \frac{1}{n} \sum_{b=0}^{B} \max_{k \in [\![1,K]\!]} m_{b,k}. \tag{4.26}$$

Les débits de transmission $R_{(b,k)}$ correspondent bien au minimum possible, comme si on stockait un mot de code spécifique pour chaque paire $(\mathbf{q}^n_{(b)}, \mathbf{y}^n_{(k)})$. En revanche, le débit de stockage implique une petite perte par rapport à l'optimal qui serait obtenu en inversant les signes max et $\sum$ dans (4.26). Cette perte est due au fait que pour chaque plan de bits, on doit stocker un mot de code correspondant à la pire information adjacente possible, qui peut-être différente pour chaque plan de bit.

4.4.5 Résultats

Nous avons appliqué la construction précédente à 6000 séquences $\mathbf{x}^n$ différentes, avec les paramètres décrits dans la Section 4.4.1. Pour cela, nous avons utilisé un code LDPC de longueur $n = 1024$, avec la construction incrémentale décrite dans la Section 4.3.3. La

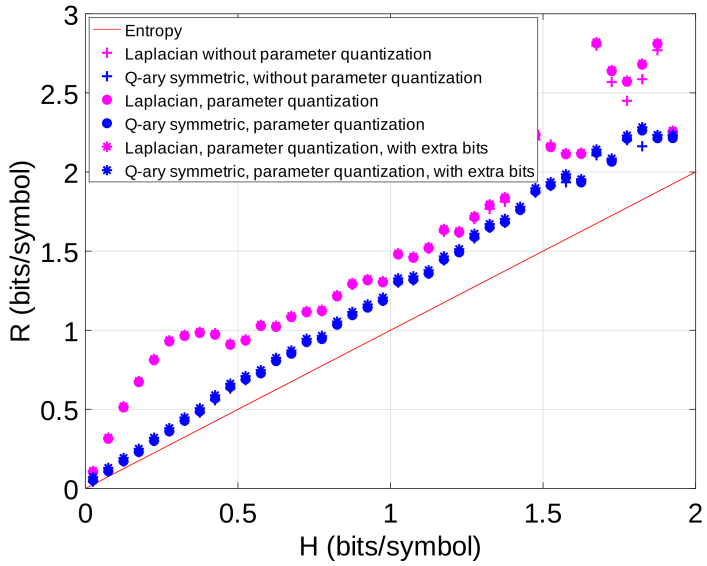

FIGURE 4.10 – Comparaison des débits de transmission pour le modèle Laplacien et le modèle q-aire.

Figure 4.10 représente les débits moyens de transmission obtenus en fonction des entropies empiriques entre les vecteurs de source et d'information adjacente. On observe que le modèle q-aire que nous avons proposé permet d'obtenir des débits plus faibles, et plus proches de l'entropie, que le modèle Laplacien de la littérature. Nous avons aussi évalué l'effet de la quantification des paramètres q_k, $Z_{k,\min}$ et $Z_{k,\max}$, du modèle dans (4.21), car ces paramètres doivent être transmis au décodeur. Nous avons observé que la quantification de ces paramètres n'induisait qu'une légère perte de performance du système de codage, ce qui permet de les représenter sur un plus petit nombre de bits.

Enfin, dans [91], nous avons aussi fourni les courbes débit-distortion du système complet (incluant la projection, la transformée, la quantification), et montré les gains obtenus vis à vis de l'approche classique avec le modèle Laplacien. Cela nous a permis de valider l'approche incrémentale proposée pour la compression d'images à 360 degrés.

4.5 Cas d'un grand nombre de sources

Dans les parties précédentes, nous avons étudié le cas d'une unique source X à compresser. Nous nous intéressons maintenant à un ensemble de sources $X_{(1)}, \cdots, X_{(J)}$ à compresser, où J peut être grand (plusieurs milliers de sources). Nous supposons qu'il

existe des dépendances statistiques entre les sources $X_{(j)}$, décrites de manière générale par une distribution de probabilité jointe $\mathbb{P}(X_{(1)}, \cdots, X_{(J)})$. Dans la suite, nous modélisons le problème d'accès à ces sources par un graphe de navigation. Nous étudions ensuite le débit de stockage de l'ensemble des sources, et le débit moyen de transmission des sources aux utilisateurs, en fonction du graphe de navigation.

4.5.1 Graphe de navigation

Dans [93], nous avons proposé de décrire les navigations possibles dans l'ensemble des sources par un graphe orienté $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, où $\mathcal{V}$ contient l'ensemble des sources $X_{(j)}$, $j \in [\![1, J]\!]$. Il existe une arrête de la source $X_{(i)}$ vers la source $X_{(j)}$ si on autorise les utilisateurs à demander la source $X_{(j)}$ juste après la source $X_{(i)}$. Nous supposons de plus que l'accès aux sources est séquentiel : l'arrête $i \to j$ signifie que la source $X_{(i)}$ servira d'information adjacente lors du décodage de la source $X_{(j)}$. En conséquence, chaque arrête est étiquetée avec l'entropie conditionnelle correspondante : l'arrête de $X_{(i)}$ vers $X_{(j)}$ aura pour label $H(X_{(j)}|X_{(i)})$.

Ce graphe de navigation permet de définir un ensemble de requêtes possibles pour les utilisateurs. On définit une requête $\mathbf{v}$ comme une suite de sources qui décrivent un chemin dans le graphe, et on note $\mathcal{V}$ l'ensemble des requêtes possibles. Chaque requête $\mathbf{v} \in \mathcal{V}$ est associée à une probabilité $\mathbb{P}(\mathbf{v})$ qui peut être calculée en fonction des probabilités de transition $p_{i,j}$ entre chaque paire de source $X_{(i)}$ vers $X_{(j)}$. Pour initialiser les requêtes, on identifie un certain nombre de sources comme étant des "points d'accès" possibles : ces sources sont alors reliées à une source virtuelle notée $X_{(0)}$. On suppose que toute navigation part de la source $X_{(0)}$, et l'arrête entre la source $X_{(0)}$ et une source $X_{(j)}$ donnée a pour label $H(X_{(j)})$.

Le graphe de navigation permet de capturer les contraintes spécifiques à une application, et d'adapter en conséquence les débits de stockage et de transmission de sources. Par exemple, pour des images à 360° (voir Section 4.1.1), les sources $X_{(j)}$ représentent les tuiles de l'image. Après réception d'une tuile $X_{(j)}$ donnée, l'utilisateur pourrait par exemple être autorisé uniquement à demander une des tuiles voisines, ce qui correspondra à un graphe de navigation relativement creux.

4.5.2 Analyse de performances

Comme dans le cas d'une seule source, on souhaite étudier le débit de stockage des sources sur le serveur, et le débit de transmission des sources du serveur aux utilisateurs. Nous définissons maintenant ces quantités dans le cas de J sources $X_{(j)}$, $j \in [\![1, J]\!]$, et d'un graphe de navigation $\mathcal{G}$ donné [93]. On peut tout d'abord exprimer le débit de stockage S de l'ensemble des sources de la manière suivante :

$$S = \frac{1}{J} \sum_{j=1}^{J} S_j, \tag{4.27}$$

où S_j représente le débit de stockage de la source $X_{(j)}$. De plus, on peut exprimer le débit de transmission $R(\mathbf{v})$ pour une requête spécifique $\mathbf{v} \in \mathcal{V}$:

$$R(\mathbf{v}) = \frac{1}{|\mathbf{v}|} \sum_{m=1}^{|v|} R_{(m-1)\to m} \tag{4.28}$$

où $|\mathbf{v}|$ est le cardinal de $\mathbf{v}$, et $R_{(m-1)\to m}$ est le débit de transmission de la m-ème source de la requête, sachant que la $(m-1)$-ème source est toujours présente au décodeur. Enfin, le débit de transmission moyen dépend de la distribution de probabilité des requêtes, et peut s'exprimer comme

$$R = \mathbb{E}\left[R(\mathbf{v})\right] = \sum_{\mathbf{v}\in\mathcal{V}} \mathbb{P}(\mathbf{v}) R(\mathbf{v}). \tag{4.29}$$

Ensuite, les valeurs des débits S_j et $R_{(m-1)\to m}$ vont dépendre de la méthode de compression utilisée, comme nous le décrivons maintenant.

Performance des schémas de compression existants

Dans la littérature, il existe deux grandes familles de méthodes de compression pour un grand nombre de sources corrélées : une approche exhaustive [10, 94], et une approche broadcast [9, 95]. Nous décrivons maintenant ces deux approches, et analysons leur performance en utilisant les définitions des débits de stockage et de transmission données dans (4.27) et (4.29).

L'approche exhaustive considérée dans [10, 94] consiste pour chaque source $X_{(j)}$ à stocker une représentation codée par information adjacente possible $X_{(i)}$, avec $i \in \mathcal{N}(j)$, où $\mathcal{N}(j)$ est l'ensemble des noeuds connectés au noeud j. Pour la source j, on obtient

donc les débits de stockage et de transmission suivants :

$$S_j = \sum_{i=1}^{|\mathcal{N}(j)|} H(X_{(j)}|X_{(i)}) \tag{4.30}$$

$$\forall i \in \mathcal{N}(j), R_{i\to j} = H(X_{(j)}|X_{(i)}). \tag{4.31}$$

Cette stratégie permet d'obtenir des débits de transmission assez faibles, mais elle induit un coût important en stockage.

A l'inverse, la stratégie broadcast employée dans [9, 95] consiste à stocker un mot de code unique pour chaque source $X(j)$, qui sera transmise entièrement à l'utilisateur quelque soit la requête. On obtient cette fois les débits suivants :

$$S_j = \max_{i\in|\mathcal{N}(j)|} H(X_{(j)}|X_{(i)}) \tag{4.32}$$

$$\forall i \in \mathcal{N}(j), R_{i\to j} = \max_{i\in|\mathcal{N}(j)|} H(X_{(j)}|X_{(i)}). \tag{4.33}$$

On voit que cette stratégie permet d'obtenir des débits de stockage plus faibles que pour l'approche exhaustive, au prix de débits de transmission plus élevés.

Performance du schéma incrémental

Dans la Section 4.2, nous avions proposé une analyse de théorie de l'information pour le cas d'une seule source X, et nous avions exprimé les débits de stockage et de transmission obtenus à partir d'un schéma de codage incrémental. Il est possible de ré-utiliser ce schéma incrémental dans le cas de plusieurs sources, car nous supposons que ces dernières sont transmises de manière séquentielle. Le schéma de codage incrémental nous permet d'obtenir les débits suivants :

$$S_j = \max_{i\in|\mathcal{N}(j)|} H(X_{(j)}|X_{(i)}) \tag{4.34}$$

$$\forall i \in \mathcal{N}(j), R_{i\to j} = H(X_{(j)}|X_{(i)}). \tag{4.35}$$

On obtient ainsi le même débit de stockage que pour le schéma broadcast, et le même débit de transmission que pour le schéma exhaustif, ce qui correspond au meilleur de chaque solution.

Pour terminer, dans [93], nous avons aussi montré que nos schémas de compression pour une seule source s'appuyant sur des codes LDPC adaptables en rendement (voir

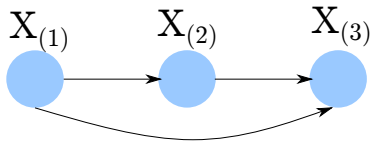

FIGURE 4.11 – Exemple de graphe de navigation entre trois sources

Section 4.3) peuvent être ré-utilisés ici et constituent une solution efficace dans le cadre d'un grand nombre de sources $X_{(j)}$.

4.5.3 Optimisation du graphe de navigation

L'analyse théorique précédente nous a permis de mettre en avant l'intérêt d'une construction incrémentale des mots de codes, y compris lorsque l'on considère un grand nombre de sources. Ceci dit, dans ce dernier cas, le débit de stockage S et le débit moyen de transmission R vont fortement dépendre du graphe de navigation $\mathcal{G}$ considéré. En effet, on peut observer dans (4.34) et (4.35) que pour la source $X_{(j)}$, les débits S_j et $R_{i\to j}$ s'expriment en fonction du voisinage $\mathcal{N}(j)$.

Prenons l'exemple d'un graphe de navigation entre trois sources $X_{(1)}, X_{(2)}, X_{(3)}$, représenté en Figure 4.11. si on suppose que $H(X_{(3)}|X_{(1)}) > H(X_{(3)}|X_{(2)})$, la suppression de l'arrête $X_{(2)}$ vers $X_{(3)}$ devrait entraîner une diminution du débit de stockage. Mais dans ce cas, pour servir une requête éventuelle $X_{(1)} \to X_{(3)}$, il deviendra nécessaire de transmettre la source $X_{(2)}$, même si elle n'est pas demandée par l'utilisateur. Cela aura alors pour effet d'augmenter le débit de transmission de cette requête.

C'est pourquoi, dans [96], nous avons proposé d'optimiser le graphe de navigation de manière à adresser un certain compromis entre le débit de stockage S et le débit moyen de transmission R. Pour adresser ce compromis, nous avons proposé d'étudier le problème d'optimisation suivant :

$$\mathcal{C}^\star = \min_{\mathcal{C}\in\mathcal{D}}(S(\mathcal{C}) + \lambda R(\mathcal{C})), \tag{4.36}$$

dans lequel l'objectif est d'obtenir un sous-graphe $\mathcal{C}^\star \subseteq \mathcal{G}$ du graphe de navigation. Ce sous-graphe devra respecter certaines contraintes, notamment de connectivité, et l'ensemble des sous-graphes $\mathcal{C}$ respectant ces contraintes est noté $\mathcal{D}$. Le paramètre λ est une donnée du problème d'optimisation (on cherchera une solution optimale pour une certaine valeur de λ), et permet d'adresser le compromis entre le débit de stockage et le débit de transmission.

Bien que le problème d'optimisation défini dans (4.36) soit formulé très simplement,

sa résolution est complexe car la variable à optimiser est un graphe, qui peut-être vu comme un objet discret de grande dimension. Dans [96], nous avons proposé une méthode d'optimisation alternée, qui s'appuie sur une étape d'identification des plus courts chemins indirects pour chaque arrête du graphe, et sur une étape de résolution d'un problème de programmation linéaire qui permet de supprimer des arrêtes du graphe de navigation. Nous avons montré l'efficacité de cette méthode pour optimiser un graphe de navigation impliquant un grand nombre de noeuds (supérieur à 1000).

4.6 Conclusion

Dans ce chapitre, nous avons étudié le problème de la compression d'un grand nombre de sources corrélées et stockées sur un serveur. Nous sommes partis d'une analyse de théorie de l'information d'une version simplifiée de ce problème, avec une unique source à compresser pour un certain nombre d'informations adjacente possibles au décodeur. En plus de nous fournir les performances limites atteignables par le système de compression, l'analyse de la théorie de l'information nous a montré qu'un schéma de codage incrémental était optimal pour ce problème. Nous avons utilisé cette idée pour construire des schémas de compression pratiques utilisant des codes LDPC adaptables en débit. Après avoir évalué la solution sur des sources synthétiques, nous avons adapté notre méthode à la compression d'images à 360 degrés. Cela nous a permis de valider notre approche dans une application réelle. De manière intéressante, nous avons ensuite réussi à généraliser l'analyse théorique ainsi le schéma pratique à un grand nombre de sources, avec une hypothèse d'accès séquentiel aux sources.

Comme extension possible, il serait intéressant d'étudier le cas où les distributions de probabilités $P(X, Y_{(k)})$ entre source et information adjacentes ne sont pas bien connues, car ces distributions sont en général difficiles à obtenir *a priori*. De plus, d'autres applications de ce problème mériteraient d'être étudiées davantage, comme la télévision interactive, ou le problème de distribution de clés quantiques [97].

Chapitre 5

Décodeurs LDPC à très faible consommation d'énergie

5.1 Introduction

En conformité avec la prédiction de Moore en 1965 [11], le nombre de transistors sur les puces électroniques a en moyenne doublé tous les deux ans. Cette évolution s'est accompagnée d'une réduction considérable des tensions d'alimentation des circuits électroniques, entraînant ainsi des progrès significatifs en termes d'efficacité énergétique. Mais réduire encore davantage les tensions d'alimentation des circuits reviendrait à se placer près du seuil des transistors. Cela créerait des effets indésirables, tels qu'une augmentation des délais dans les circuits et un taux d'erreur plus important dans les portes logiques comme dans les mémoires [98]. Cependant, les gains potentiels ne sont pas négligeables, car la consommation d'énergie est quadratique avec la tension d'alimentation.

Pour pouvoir exploiter ces gains potentiels en énergie, une solution intéressante serait de concevoir des méthodes de calcul robustes aux effets indésirables, plutôt que de s'appuyer uniquement sur des solutions purement matérielles. La plupart des méthodes de traitement de signal et de machine learning étant conçues pour gérer les effets du bruit, il est en effet raisonnable de considérer qu'elles pourraient aussi être adaptées pour gérer également le bruit provenant du circuit.

5.1.1 Types d'erreurs

Les erreurs introduites dans le décodeur peuvent être de différentes natures, que nous présentons maintenant brièvement. Tout d'abord, les procédés de fabrication plus complexes des dernières technologies CMOS peuvent entraîner des erreurs permanentes, avec des cellules mémoires ou des portes logiques qui seront toujours défaillantes. On parle d'erreurs de type "stuck-at", car les unités concernées vont toujours renvoyer la même va-

leur [99, 100]. Mais ces erreurs ne sont pas directement liées à la consommation d'énergie des systèmes, et nous ne les considérerons donc pas ici.

A l'inverse, les erreurs dites transientes apparaissent de temps en temps et aléatoirement dans les mémoires et dans les unités de calcul [101]. Dans les mémoires de type DRAM, le niveau logique 1 est représenté par une charge électrique, tandis que le niveau 0 est représenté par l'absence de charge électrique [102]. Dans les mémoires plus récentes de type ReRAM, les niveaux logiques 1 et 0 sont représentés par des valeurs de résistances particulières pour les composants résistifs utilisés dans la mémoire [103]. Pour réduire la consommation d'énergie de ces mémoires, on peut par exemple réduire la charge électrique ou la valeur de la résistance correspondant au niveau 1. Mais cela aura pour effet d'augmenter les erreurs, qui peuvent dans ce cas être modélisé comme un bruit introduit dans les cellules mémoires.

En sortie des portes logiques, un bruit similaire peut apparaître lorsque l'on se place à une tension d'alimentation près du seuil [98] car alors l'écart de tension est extrêmement réduit entre le niveau logique 1 et le niveau logique 0. De plus, les sorties des portes logiques peuvent mettre plus longtemps à converger [104]. Dans ce cas de figure, si on décide d'éviter une approche pire cas coûteuse en consommation d'énergie [100], il sera nécessaire de concevoir des architectures de calcul supportant ces erreurs de timing.

5.1.2 Codes correcteurs d'erreurs sur circuits bruités

Dans ce chapitre, nous allons nous concentrer sur le cas particulier des algorithmes de décodage de codes correcteurs d'erreurs implémentés sur des circuits bruités. L'étude de ces algorithmes présente un double intérêt. En premier lieu, Les systèmes de télécommunications modernes font tous appel à des codes correcteurs d'erreurs qui permettent de garantir une excellente fiabilité dans la transmission des messages, sans augmenter la puissance d'émission. Mais cela se fait au prix d'une consommation d'énergie importante au récepteur, pour alimenter les circuits qui implémentent les algorithmes de décodage.

Deuxièmement, afin d'améliorer la robustesse aux erreurs dans d'autres algorithmes de calcul, de traitement du signal, ou d'apprentissage automatique, il serait intéressant d'y intégrer des mécanismes de correction d'erreurs. Cela implique d'utiliser des circuits de décodage capables de fonctionner efficacement sur des circuits bruités. Par conséquent, il est indispensable de concevoir au préalable des algorithmes de décodage à la fois robustes et économes en énergie, pour rendre envisageable l'idée de sous-alimenter les circuits.

5.1.3 Décodeurs LDPC sur circuits bruités

Dans ce chapitre, nous allons étudier le cas particulier des codes LDPC, qui sont utilisés dans un grand nombre de standards de télécommunications (5G, DVB-S2, etc.). La problèmatique générale de la conception de décodeurs LDPC à faible consommation d'énergie est difficile, car très transverse.

Pour l'aborder, nous allons tout d'abord analyser théoriquement l'effet des erreurs dans les décodeurs LDPC, pour évaluer leur robustesse et être en capacité de concevoir des codes et décodeurs LDPC résilients aux erreurs. Cependant, pour que l'étude soit pertinente, il est nécessaire de considérer des modèles d'erreurs à la fois réalistes *et* tractables du point de vue de l'analyse théorique, ce qui constitue toujours un équilibre délicat.

Une difficulté supplémentaire est que les modèles d'erreur (type d'erreurs, endroit où elles sont introduites, dans quelles proportions, etc.), tout comme la consommation d'énergie du décodeur, vont dépendre fortement de l'architecture matérielle considérée. Même si des tentatives de modèles génériques de consommation d'énergie ont été proposés, ils sont peu réalistes [106], ne concernent que les mémoires [107], ou dépendent implicitement de paramètres spécifiques à l'architecture, comme la longueur des fils sur le circuit, ou le nombre d'emplacements mémoires [108]. On peut donc difficilement s'affranchir de fixer une architecture spécifique, pour l'étudier et optimiser sa consommation d'énergie vis à vis d'un certain nombre de leviers : la structure du code, le choix du décodeur, les paramètres spécifiques à l'architecture.

C'est pourquoi, dans ce chapitre, nous avons privilégié l'approche suivante, en deux étapes :

1. **Analyse théorique de l'effet des erreurs dans les décodeurs** : nous avons utilisé des méthodes d'évolution de densité pour prédire l'effet du bruit dans les décodeurs LDPC, pour des modèles d'erreur plus réalistes. Nous avons en particulier considéré deux modèles d'erreurs importants : un modèle avec des erreurs asymétriques, et un modèle représentant des erreurs de timing.

2. **Etude et optimisation de la consommation d'énergie d'une architecture spécifique** : nous avons proposé une architecture matérielle de décodeurs LDPC fortement parallèle, et optimisé sa consommation d'énergie en jouant sur plusieurs leviers : le code utilisé, les paramètres du décodeur, le niveau de bruit, etc.

Nous présentons maintenant les contributions associées à chacun de ces deux axes.

5.2 Effet des erreurs dans les décodeurs LDPC

Dans cette partie, nous étudions l'effet des erreurs dans les décodeurs LDPC. Nous considérons que ces erreurs peuvent apparaître dans les mémoires et dans les portes logiques, et sont de différentes natures que nous détaillerons dans la suite.

5.2.1 État de l'art

Dans [109], une nouvelle méthode d'évolution de densité (voir Section 3.4) a été introduite pour prendre en compte les erreurs dans les décodeurs LDPC. Ce premier travail, qui se concentrait sur les décodeurs Gallager A et BP, en a appelé de nombreux autres, principalement sur les décodeurs Gallager B [110, 111] et MS quantifié [112, 113, 114]. Tous ces travaux concluent que même si le bruit implique une dégradation de performance, les décodeurs LDPC sont de manière inhérente relativement robustes aux erreurs introduites dans les mémoires et dans les portes logiques, ce qui est tout à fait cohérent avec le fait qu'ils sont conçus à l'origine pour corriger des erreurs. Ils permettent aussi d'identifier à partir de quel niveau d'erreur les performances commencent à se dégrader sérieusement, ce qui permettra de dimensionner le système en conséquence.

Certains travaux montrent même que le bruit peut aider le décodage, par exemple dans des algorithmes de type Bit-Flipping [115], ou MS quantifié [116, 117]. Ces résultats sont cependant à interpréter avec précaution, car ils impliquent plutôt des systèmes d'injection de bruit maîtrisés et avec un niveau fixé.

Cependant, les travaux précédents considèrent le plus souvent des modèles de bruit relativement simples, à la fois i.i.d. et préservant la symétrie du décodeur. Ces hypothèses simplifient grandement l'évolution de densité, mais sont peu réalistes en pratique. Ceci dit, si on souhaite retirer ces hypothèses, on peut tout d'abord préciser que considérer des erreurs non-identiquement distribuées ne représente pas une grande difficulté, tant que l'hypothèse de symétrie est préservée. C'est ce l'on fait par exemple lorsque l'on considère une évolution de densité pour des codes décrits par des protographes [84, 118]. En revanche, considérer des erreurs non-indépendantes rend l'analyse très complexe [119, 120].

Dans la suite, après avoir décrit comment prédire la performance des décodeurs bruités à partir de l'évolution de densité, je présente mes contributions liées à la prise en compte de modèles de bruit plus réalistes. Je me concentre sur le cas d'erreurs asymétriques, qui oblige à retirer l'hypothèse du mot de code tout à zéro, et sur le cas des erreurs de timing.

Ces erreurs introduisent des dépendances statistiques dans les messages échangés dans le décodeur, et c'est pourquoi nous proposons une analyse alternative à l'évolution de densité.

5.2.2 Évolution de densité pour les décodeurs bruités

Dans cette partie, nous reprenons les grandes étapes de la méthode d'évolution de densité décrite en Section 3.4, et décrivons comment elle est modifiée lorsque l'on considère que des erreurs sont introduites dans les opérations de calcul.

Seuil opérationnel

L'évolution de densité permet de prédire la probabilité d'erreur $P_e^{(\ell)}$ du décodeur. Le seuil d'un ensemble de codes est ensuite estimé comme le pire paramètre du canal pour lequel $P_e^{(\ell)}$ tend vers 0 pour ℓ assez grand et quand N tend vers l'infini. Mais cette condition n'est plus applicable dans le cas d'un décodeur bruité, pour lequel on peut montrer que $P_e^{(\ell)}$ possède une borne inférieure $\overline{P}$ strictement supérieure à 0, qui dépend du niveau de bruit dans le décodeur [121].

Dans une définition alternative appelée seuil "opérationnel", on se fixe un paramètre $\epsilon \geq \overline{P}$ et on détermine le pire paramètre du canal pour lequel la probabilité d'erreur passe en dessous de ce seuil pour un nombre d'itérations ℓ assez grand [109]. Pour un BSC, le seuil opérationnel est donc défini comme

$$\overline{p} = \max p \quad \text{tel que} \quad P_e^{(\ell)}(p) < \epsilon \tag{5.1}$$

où ϵ est un paramètre à fixer (typiquement 10^{-6} ou en dessous). De même, pour un canal BiAWGN, le seuil opérationnel est défini comme

$$\overline{\mathrm{SNR}} = \min \mathrm{SNR} \quad \text{tel que} \quad P_e^{(\ell)}(\mathrm{SNR}) < \epsilon, \tag{5.2}$$

où SNR est le rapport signal-à-bruit du canal.

Notons que d'autres définitions ont été proposées, comme le seuil fonctionnel [112, 122], qui caractérise le point inflexion dans la courbe de probabilité d'erreur, entre une zone de faible taux d'erreur et une zone de fort taux d'erreur. En pratique, on utilise le plus souvent la définition du seuil opérationnel, bien plus simple à calculer.

Modèle d'erreurs

Dans la Section 3.4, nous avons décrit l'évolution de densité usuelle pour des décodeurs quantifiés, impliquant des messages $u^{(\ell)}$ (calculés par les NC) et $t^{(\ell)}$ (calculés par les NV), prenant leurs valeurs dans le même alphabet discret $\mathcal{M} = [\![-q, +q]\!]$. L'évolution de densité permet de calculer les probabilités $\mathbb{P}(u^{(\ell)})$ et $\mathbb{P}(t^{(\ell)})$ pour les itérations successives $\ell \in [\![1, L]\!]$.

Dans cette partie, nous supposons, comme dans la majorité des travaux existants, que les erreurs sont introduites en sortie du calcul des fonctions aux NV et aux NC. On note $\tilde{u}^{(\ell)}$ et $\tilde{t}^{(\ell)}$ les versions bruitées de $u^{(\ell)}$ et $t^{(\ell)}$, respectivement, où $\tilde{u}^{(\ell)}$ et $\tilde{t}^{(\ell)}$ prennent leurs valeurs dans le même alphabet quantifié $\mathcal{M}$. On peut alors décrire le modèle d'erreurs par une matrice de transitions Π de dimension $(2q+1) \times (2q+1)$, telle que [122]

$$\Pi_{i,j} = P(\tilde{u}^{(\ell)} = j | u^{(\ell)} = i) \tag{5.3}$$

pour tout $i, j \in \mathcal{M}$. Dans la suite, nous allons considérer pour simplifier que la matrice de transition est la même pour les NV et les NC, et qu'elle ne varie pas avec l'itération ℓ. La généralisation a des cas plus complexes est triviale.

De plus, la majorité des travaux considèrent des modèles d'erreur symétriques, qui permettent de préserver l'hypothèse du mot de code tout à zéro [109, 122, 111, 112, 113, 114]. Pour ces modèles symétriques, on a $\Pi_{i,j} = \Pi_{-i,-j}$ [122]. Par exemple, si le décodeur échange des messages binaires -1 et 1, on aura $P(\tilde{u}^{(\ell)} = 1 | u^{(\ell)} = -1) = P(\tilde{u}^{(\ell)} = -1 | u^{(\ell)} = 1)$. C'est ce cas symétrique que je décris maintenant.

Mise à jour de l'évolution de densité pour des décodeurs bruités

Lorsque l'on souhaite effectuer l'évolution de densité pour des décodeurs bruités quantifiés, le calcul des probabilités $\mathbb{P}(u^{(\ell)})$ et $\mathbb{P}(t^{(\ell)})$ peut toujours s'effectuer à l'aide des équations (3.15) et (3.16) obtenues dans le cas sans bruit, à ceci près que les probabilités d'entrée utilisées sont maintenant $\mathbb{P}(\tilde{t}^{(\ell)})$ et $\mathbb{P}(\tilde{u}^{(\ell)})$. Ensuite, pour calculer les probabilités mises à jour des messages $\tilde{u}^{(\ell+1)}$ et $\tilde{t}^{(\ell+1)}$, on peut utiliser les relations matricielles suivantes [122]

$$\mathbf{P}_{\tilde{\mathbf{t}}}^{(\ell)} = \Pi \mathbf{P}_{\mathbf{t}}^{(\ell)}, \tag{5.4}$$

$$\mathbf{P}_{\tilde{\mathbf{u}}}^{(\ell)} = \Pi \mathbf{P}_{\mathbf{u}}^{(\ell)}. \tag{5.5}$$

Dans ces expressions, $\mathbf{P}_{\mathbf{u}}^{(\ell)}$ et $\mathbf{P}_{\tilde{\mathbf{u}}}^{(\ell)}$ représentent les vecteurs de dimension $(2q+1)$ qui contiennent l'ensemble des probabilités $\mathbb{P}(u^{(\ell)}), \mathbb{P}(\tilde{u}^{(\ell)})$ pour tout $u^{(\ell)}, \tilde{u}^{(\ell)} \in \mathcal{M}$ à l'itération ℓ. De même, $\mathbf{P}_{\mathbf{t}}^{(\ell)}$ et $\mathbf{P}_{\tilde{\mathbf{t}}}^{(\ell)}$ représentent les vecteurs qui contiennent l'ensemble des probabilités $\mathbb{P}(t^{(\ell)})$ et $\mathbb{P}(\tilde{t}^{(\ell)})$ pour tout $t^{(\ell)}, \tilde{t}^{(\ell)} \in \mathcal{M}$.

On peut ensuite appliquer les méthodes décrites dans les sections précédentes pour calculer le seuil du décodeur, en fonction des caractéristiques du canal et du niveau de bruit décrit par la matrice Π. De plus, la méthode d'évolution de densité à longueur finie décrite dans la Section 3.4.2 peut aussi s'appliquer aux décodeurs bruités, en utilisant la probabilité d'erreur $\tilde{P}_e^{(\ell)}$ du décodeur bruité, au lieu de $P_e^{(\ell)}$ dans (3.18).

5.2.3 Modèles d'erreurs asymétriques

En pratique, l'hypothèse de modèles d'erreurs symétriques n'est pas toujours vérifiée. Par exemple, on sait que pour des mémoires de type eDRAM [102] et SRAM [123], la probabilité d'erreur dépend de la valeur du bit. C'est pourquoi, dans [124], nous avons développé une nouvelle méthode d'évolution de densité permettant de prendre en compte des modèles d'erreurs asymétriques. Dans ce cas, la matrice de transition Π peut-être n'importe quelle matrice qui représente une distribution de probabilité. La difficulté majeure réside dans le fait que l'on ne peut plus considérer l'hypothèse du mot de code tout à zéro, car dans ce cas, la probabilité d'erreur du décodeur est différente pour un bit à 0 ou à 1 dans le mot de code.

Expression de la probabilité d'erreur

Nous supposons donc maintenant qu'un mot de code quelconque $\mathbf{x}$ a été transmis sur le canal. Nous allons conditionner toutes les probabilités impliquées dans l'évolution de densité à la valeur binaire $x_v \in \{0,1\}$ associée au NV v impliqué dans le message en cours de calcul. A chaque itération, nous allons donc devoir évaluer quatre distributions de probabilités $\mathbb{P}_0(\tilde{t}^{(\ell)})$, $\mathbb{P}_1(\tilde{t}^{(\ell)})$ et $\mathbb{P}_0(\tilde{u}^{(\ell)})$, $\mathbb{P}_1(\tilde{u}^{(\ell)})$. Ensuite, on peut exprimer la probabilité d'erreur du décodeur asymétrique de la manière suivante :

$$\tilde{P}_e^{(\ell)} = \sum_{t^{(\ell)}<0} \left(\frac{1}{2}\mathbb{P}_0(\tilde{t}^{(\ell)}) + \frac{1}{2}\mathbb{P}_1(\tilde{t}^{(\ell)}) \right) + \frac{1}{2}\mathbb{P}(0). \tag{5.6}$$

Cette expression est obtenue en marginalisant vis à vis des valeurs $x_v = 0$ et $x_v = 1$. De plus, les facteurs $1/2$ viennent du fait que les bits du mot de code $\mathbf{x}$ sont équiprobables.

Mise à jour de l'évolution de densité pour des modèles d'erreurs asymétriques

Le fait de devoir considérer les deux valeurs possibles $x = 0$ et $x = 1$ a une incidence importante sur le calcul des probabilités dans l'évolution de densité. Dans notre exemple du décodeur quantifié, on a ainsi pour les probabilités des messages en sortie des NV,

$$\mathbb{P}_x(t^{(\ell)}) = \sum_{\mathbf{u}^{(\ell)} \in \mathcal{M}^{d_v} : \Psi_v(\mathbf{u}^{(\ell)}) = t^{(\ell)}} \mathbb{P}_x(t^{(0)}) \prod_{d=1}^{d_v - 1} \mathbb{P}_x(u_d^{(\ell)}), \tag{5.7}$$

et pour les probabilités des messages en sortie des NC,

$$\mathbb{P}_0(u^{(\ell)}) = \left(\frac{1}{2}\right)^{d_c - 2} \sum_{v=0,\ \text{even}}^{d_c - 1} \binom{d_c - 1}{v} \sum_{\mathbf{t}^{(\ell)} \in \mathcal{M}^{d_c - 1} : \Psi_c(\mathbf{t}^{(\ell-1)}) = u^{(\ell)}} \prod_{d=1}^{v} \mathbb{P}_1(t_d^{(\ell)}) \prod_{d=v+1}^{d_c - 1} \mathbb{P}_0(t_d^{(\ell)}) \tag{5.8}$$

$$\mathbb{P}_1(u^{(\ell)}) = \left(\frac{1}{2}\right)^{d_c - 2} \sum_{v=1,\ \text{odd}}^{d_c - 1} \binom{d_c - 1}{v} \sum_{\mathbf{t}^{(\ell)} \in \mathcal{M}^{d_c - 1} : \Psi_c(\mathbf{t}^{(\ell-1)}) = u^{(\ell)}} \prod_{d=1}^{v} \mathbb{P}_1(t_d^{(\ell)}) \prod_{d=v+1}^{d_c - 1} \mathbb{P}_0(t_d^{(\ell)}). \tag{5.9}$$

Ensuite, les relations matricielles (5.4) et (5.5) restent valides pour la prise en compte du bruit en sortie des NV et des NC.

Paramètres asymétriques dans les décodeurs

La plupart des décodeurs LDPC dépendent de paramètres qu'il est nécessaire d'optimiser pour améliorer la performance de décodage. Par exemple, un décodeur MS sera augmenté avec un paramètre particulier appelé offset, qui est en général fixé pour tout le décodage ou par itération, et dont la valeur ne dépend pas des messages échangés dans le décodeur.

Néanmoins, dans [124], nous avons proposé de considérer des paramètres asymétriques, c'est à dire variant en fonction du signe des messages, pour pouvoir corriger l'effet des erreurs asymétriques introduites dans le décodeur. Ainsi, dans le cas d'un décodeur MS, nous avons considéré deux paramètres d'offsets différents γ_p et γ_n, et utilisé le mapping suivant au niveau des NC :

$$u_{d_c}^{(\ell)} = \begin{cases} \left(\prod_{d=1}^{d_c-1} \operatorname{sgn}\left(t_d^{(\ell)}\right)\right) \max\left(\min_d \left|t_d^{(\ell)}\right| - \gamma_p, 0\right) & \text{si } \prod_{d=1}^{d_c-1} \operatorname{sgn}\left(t_d^{(\ell)}\right) > 0, \\ \left(\prod_{d=1}^{d_c-1} \operatorname{sgn}\left(t_d^{(\ell)}\right)\right) \max\left(\min_d \left|t_d^{(\ell)}\right| - \gamma_n, 0\right) & \text{si } \prod_{d=1}^{d_c-1} \operatorname{sgn}\left(t_d^{(\ell)}\right) < 0, \\ 0 & \text{sinon.} \end{cases} \tag{5.10}$$

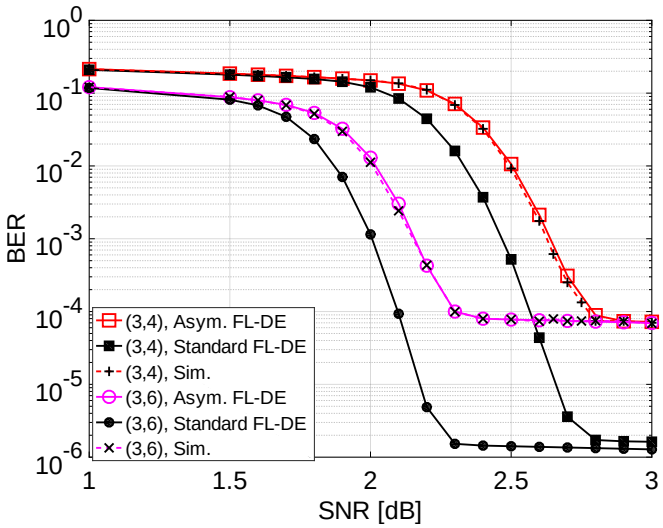

FIGURE 5.1 – Pour un décodeur MS quantifié, comparaison des évolutions de densité symétrique et asymétrique à longueur finie, avec les résultats de simulations de Monte Carlo.

A l'aide d'une évolution de densité prenant en compte ces offset, nous avons montré que dans le cas de modèles d'erreurs asymétriques, utiliser des paramètres asymétriques permettait d'améliorer la performance de manière significative.

Résultats numériques

Nous présentons maintenant quelques résultats de simulations pour le décodeur MS quantifié appliqué après un canal AWGN. Dans cette partie, par simplicité, nous considérons uniquement des codes réguliers, même si l'analyse s'appliquerait également à des codes irréguliers ou décrits par des protographes.

La Figure 5.1 représente le BER en fonction du SNR, pour des codes réguliers $(3, 4)$ et $(3, 6)$ de longueur $N = 10000$. Les courbes sont obtenues à partir de l'évolution de densité symétrique classique, à partir de l'évolution de densité asymétrique proposée, et à partir de simulations de Monte Carlo. Pour les courbes d'évolution de densité, nous avons appliqué la méthode décrite par l'équation (3.18), qui permet de prédire la performance à longueur finie (FL-DE). On observe que les courbes d'évolution de densité asymétrique sont superposées aux résultats de simulations de Monte Carlo. En revanche, les courbes d'évolution de densité symétrique sont assez éloignées des résultats de simulations, ce qui était attendu. Ces résultats permettent de confirmer que la méthode d'évolution de

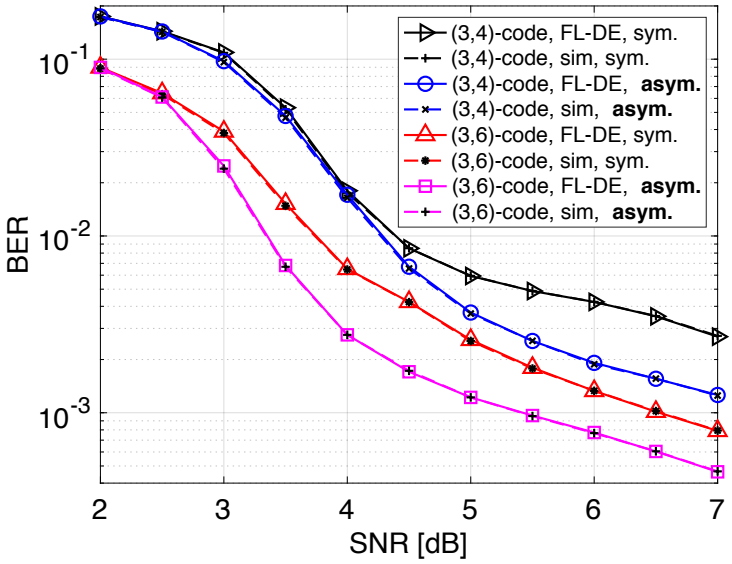

FIGURE 5.2 – Pour un décodeur MS quantifié, performance des décodeurs avec offsets symétriques et asymétriques, évalués par la méthode d'évolution de densité asymétrique et par des simulations de Monte Carlo.

densité proposée permet de prédire avec une très grande précision la performance des décodeurs asymétriques.

La Figure 5.2 représente le BER en fonction du SNR pour des codes réguliers $(3, 4)$ et $(3, 6)$ de longueur $N = 10000$, avec pour objectif d'évaluer la performance de l'offset asymétrique décrit dans l'équation (5.10). A partir de cette figure, on observe à nouveau que la méthode d'évolution de densité asymétrique permet de prédire avec une grande précision les résultats de simulations de Monte Carlo. On observe également que le paramètre d'offset asymétrique permet d'obtenir un gain conséquent en terme de BER par rapport à l'offset symétrique. Cela confirme l'intérêt d'utiliser des paramètres asymétriques lorsque le modèle d'erreur est asymétrique.

5.2.4 Erreurs de timing

Enfin, dans [125], nous avons considéré un décodeur de type Gallager B, et étudié les erreurs de timing, commises lorsque le circuit prend une décision sur une valeur en sortie du porte logique, alors que le signal électrique ne s'est pas encore stabilisé.

Modèle d'erreurs

Concrètement, dans un décodeur Gallager B, les messages $t^{(\ell)}$ et $u^{(\ell)}$ prennent des valeurs binaires 0 ou 1. Les erreurs de timing produisent des messages potentiellement erronés de la forme [104, 126]

$$\tilde{t}^{(\ell)} = \overline{D}^{(\ell)}.t^{(\ell)} \oplus D^{(\ell)}.t^{(\ell-1)}, \tag{5.11}$$

où $D^{(\ell)} \in \{0,1\}$ avec $\mathbb{P}(D^{(\ell)} = 1) = p$. Cela signifie que $\tilde{t}^{(\ell)}$ est égal à la valeur correcte $t^{(\ell)}$ avec une probabilité $1-p$, et est égale à la valeur précédente $t^{(\ell-1)}$ avec une probabilité p. Ce type d'erreur introduit donc des dépendances statistiques entre les messages aux itérations successives. Ces dépendances rendent l'évolution de densité beaucoup trop complexe à évaluer [119]. Nous avons donc proposé l'analyse alternative suivante.

Analyse de l'effet des erreurs de timing

Comme dans les travaux qui étudient le scheduling en série [127], nous avons considéré la notion de graphe de décodage. Concrètement, pour une arrête $e = (v, c)$ du graphe de Tanner, le graphe de décodage $\mathcal{N}_e^{(\ell)}$ dans le décodeur sans erreur à l'itération ℓ inclut tous les noeuds impliqués dans la valeur du message de c vers v à l'itération ℓ. Pour un décodeur sans erreur, ce graphe inclus donc tous les NV et NC à distance strictement inférieure à $2\ell - 1$ de v. On peut définir également le graphe de décodage $\widetilde{\mathcal{N}}_e^{(\ell)}$ pour le décodeur contenant des erreurs de timing. Cette fois, il s'agit d'un graphe aléatoire, qui dépendra des erreurs introduites par le modèle (5.11). En effet, le modèle décrit par (5.11) introduit simplement des retards dans la prise en compte des messages dans le décodeur. Dans [125], nous avons montré la relation suivante entre les graphes de décodage :

$$\mathcal{N}_e^{(\ell+1)} \subseteq \widetilde{\mathcal{N}}_e^{(3\ell)} \subseteq \mathcal{N}_e^{(3\ell)}. \tag{5.12}$$

Cette relation entre les graphes de décodage permet de montrer que si pour le décodeur sans erreur, la probabilité d'erreur $P_e^{(\ell)}$ converge avec ℓ, alors

$$\widetilde{P}_e^{(+\infty)} = P_e^{(+\infty)} \tag{5.13}$$

En conclusion, les erreurs de timing augmentent simplement le nombre d'itérations nécessaires au décodage, mais elles ne changent pas la performance en terme de BER ou FER

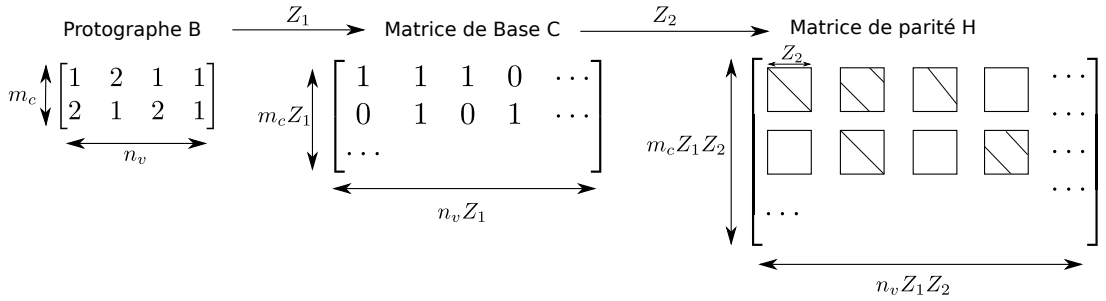

FIGURE 5.3 – Construction d'une matrice LDPC QC, en partant d'un protographe de dimension $m_c \times n_v$. On effectue ensuite deux extensions, de facteurs respectifs Z_1 et Z_2, pour obtenir une matrice H de dimension $m \times n$

quand le nombre d'itérations est assez grand.

Ces résultats ont été démontrés dans le cas du décodeur Gallager B, mais l'analyse pourrait s'étendre facilement à d'autres types de décodeur comme le BP ou le MS. Ceci dit, une hypothèse implicite de ce travail est que chaque NV ou NC est implémenté à partir d'un processeur unique, ce qui n'est pas toujours le cas en pratique. Mais si cette hypothèse n'est pas vérifiée, on peut se ramener à des modèles d'erreurs décrits par des taux d'erreurs, comme considérés plus tôt dans le chapitre.

5.3 Architecture matérielle

Dans cette partie, je présente une architecture spécifique, fortement parallèle, que nous avons proposé dans [128]. Dans les parties suivantes, nous étudierons la consommation d'énergie de cette architecture.

Cette architecture utilise un décodeur offset MS quantifié, et est adaptée au décodage de codes LDPC quasi-cycliques (QC) construits de la manière suivante. En reprenant les notations de la Section 3.2.3, on considère un protographe B de dimension $m_c \times n_v$, que l'on étend en une matrice de base C de dimension $(Z_1 m_c) \times (Z_1 m_v)$. On appelle les NC contenus dans la matrice de base C les C-CN, et les NV contenus dans C les C-VN. Puis chaque composante de C est remplacée par une matrice circulante de dimension $Z_2 \times Z_2$, pour obtenir une matrice H de dimension $m \times n$, avec $m = m_c Z_1 Z_2$ et $n = n_v Z_1 Z_2$. Cette construction est synthétisée sur la Figure 5.3.

5.3.1 Scheduling et parallélisme

Il existe plusieurs manières d'ordonnancer le décodage d'un code LDPC. Dans un scheduling standard dit parallèle, tous les NC sont évalués en parallèle, et tous les NV sont évalués en parallèle [127]. A l'opposé, dans un scheduling série, chaque NC va être évalué l'un après l'autre. Ainsi, pour un NC c donné, on évalue tout d'abord tous les messages des NV connectés à c, puis on évalue les messages sortants de c. Grâce à cela, un NC c' qui sera évalué après c pourra bénéficier d'une partie des messages qui auront déjà été mis à jour. Dans [127], il est montré qu'un décodeur utilisant un scheduling série a besoin de deux fois moins d'itérations pour converger, ce qui représente un gain très appréciable. Cependant, il devient dans ce cas beaucoup plus complexe de tirer partie d'une architecture parallèle, ce qui introduit une latence importante dans le décodage. En effet, dans cette configuration, chaque itération est évaluée en N cycles d'horloge.

C'est pourquoi des approches hybrides ont été proposées [129, 130], avec par exemple des scheduling série par groupe, où à chaque instant, un nombre J de NC sont évalués en parallèle. Dans ce cas, il faut s'assurer que les J NC n'ont pas de NV en commun, car cela créerais des conflits dans le calcul des messages. Dans notre construction précédente, il est par exemple facile de créer $m_c Z_1$ groupes de NC, contenant chacun Z_2 NC, que l'on peut évaluer en parallèle. En effet, grâce à la construction QC, on sait que ces NC ne partagent pas de NV en commun. Dans ce cas, une itération sera évaluée en $m_c Z_1$ cycles d'horloge.

Dans [128, 131], nous avons proposé une architecture permettant d'augmenter encore le degré de parallélisme, et donc de diminuer la latence de décodage. Nous présentons maintenant quelques aspects saillants de cette architecture.

5.3.2 Description de l'architecture

Dans l'architecture que nous avons proposé dans [128, 131], et qui est représentée sur la Figure 5.4, les Z_2 CN associés à un même C-CN sont évalués en parallèle. De plus, on utilise un pipeline de profondeur D_P, pour l'évaluation à chaque cycle d'horloge des C-CN entre les positions $jD_P + 1$ et $(j+1)D_P$, pour $j \in \left[\!\left[1, \frac{mcZ_1}{D_P} - 1\right]\!\right]$. Donc si on suppose que le pipeline est idéal, une itération sera évaluée en $m_c Z_1 / D_P$ cycles d'horloge.

Pour le calcul des messages dans le décodeur, on maintient pour chaque NV une somme totale des messages entrants dans le noeud, notée $t_v^{(\ell)}$. Quand le NC c dans le pipeline est

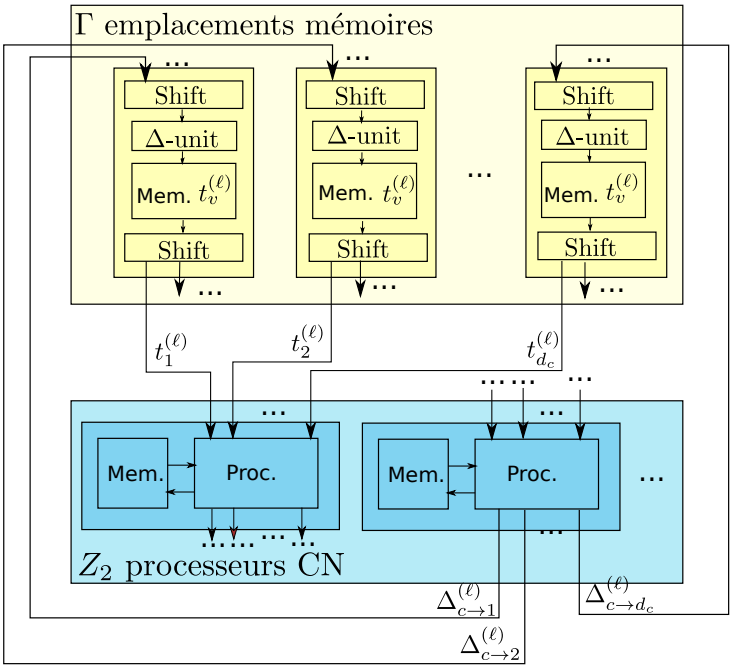

FIGURE 5.4 – Schéma de l'architecture matérielle proposée dans [128, 131] pour un décodeur MS quantifié et des codes LDPC QC.

évalué, les messages suivants sont calculés pour tout $v \in \mathcal{V}_c$:

$$u_{c\to v}^{(\ell)} = \left(\prod_{v' \in \mathcal{V}_c \setminus v} \operatorname{sgn}\left(t_v^{(\ell)} - u_{c\to v'}^{(\ell-1)}\right) \right) \max\left(\min_{v' \in \mathcal{V}_c \setminus v} |t_v^{(\ell)} - u_{c\to v'}^{(\ell-1)}| - \lambda, 0 \right) \tag{5.14}$$

$$\Delta_{c\to v}^{(\ell)} = u_{c\to v}^{(\ell)} - u_{c\to v}^{(\ell-1)} \tag{5.15}$$

La quantité $\Delta_{c\to v}^{(\ell)}$ contient la différence entre le nouveau message $u_{c\to v}^{(\ell)}$ qui vient d'être calculé, et le message précédent $u_{c\to v}^{(\ell-1)}$. Ensuite, lorsque les D_P NV dans le pipeline ont été évalués, on met à jour les sommes totales pour les D_P NC c qui ont été mis à jour, de la manière suivante :

$$t_v^{(\ell)} = t_v^{(\ell)} + \Delta_{c\to v}^{(\ell)}. \tag{5.16}$$

Ainsi, la somme totale $t_v^{(\ell)}$ peut s'exprimer comme

$$t_v^{(\ell)} = u_v^{(0)} + \sum_{c'} u_{c'\to v}^{(\ell)} + \sum_{c''} u_{c''\to v}^{(\ell-1)} \tag{5.17}$$

où $c', c'' \in \mathcal{C}_v$, mais c' correspond aux NC déjà évalués à l'itération ℓ, tandis que c'' correspond aux NC qui n'ont pas encore été évalués à l'itération ℓ.

5.3.3 Choix des NC dans le même pipeline

Le mécanisme qui calcule les quantités $\Delta_{c\to v}^{(\ell)}$ est appelé Δ-update. Ce mécanisme permet d'éviter les conflits d'accès à la mémoire, car il stocke séparément la différence entre le message courant et le message précédent. Les $\Delta_{c\to v}^{(\ell)}$ sont ensuite utilisés uniquement à la fin du pipeline pour mettre à jour les sommes totales $t_v^{(\ell)}$. En conséquence, si deux NC évalués dans le même pipeline sont connectés à un même NV, le deuxième ne pourra pas bénéficier de la mise à jour effectuée par le premier. C'est pourquoi, dans l'idéal, on souhaite que la condition suivante soit vérifiée :

$$\forall j \in \left[\!\left[1, \frac{m_c Z_1}{D_P} - 1\right]\!\right], \forall c, c' \in [\![jD_P + 1, (j+1)D_P]\!], \quad \mathcal{V}_c \bigcap \mathcal{V}'_c = \emptyset \tag{5.18}$$

Mais la condition précédente est difficile à satisfaire en pratique. C'est pourquoi, dans [128], nous avons proposé une méthode d'optimisation pour choisir l'ordre d'évaluation des NC, de manière à minimiser le cardinal de l'ensemble $\mathcal{V}_c \bigcap \mathcal{V}'_c$ pour tous les NC c, c' dans le même pipeline. La méthode proposée consiste en une approche gloutonne, qui ajoute un NC après l'autre dans la liste ordonnée pour chaque pipeline, de manière à minimiser localement la quantité précédente.

5.3.4 Construction de codes pour minimiser le nombre d'emplacements mémoire

L'architecture précédente nécessite Z_2 processeurs pour le calcul des NC, ainsi que $\Gamma \geq d_{c,\max}$ emplacements mémoires. Quand on évalue un NC c de degré d_c, on a besoin d'accéder simultanément à d_c emplacements mémoires qui doivent donc être indépendants pour éviter les conflits d'accès. On note $\mathcal{M}_\gamma$ l'ensemble des NV qui sont associés à l'emplacement mémoire γ, avec $\gamma \in [\![1, \Gamma]\!]$. Ainsi, il faut que la condition suivante soit vérifiée :

$$\forall \gamma \in [\![1, \Gamma]\!], \forall v, v' \in \mathcal{M}_\gamma, \quad \mathcal{C}_v \bigcap \mathcal{C}_{v'} = \emptyset \tag{5.19}$$

C'est à dire que les NV alloués à un emplacement mémoire particulier ne doivent partager aucun NC en commun.

Il est donc essentiel de dimensionner correctement le nombre Γ d'emplacements mémoires, de manière à respecter la condition précédente. Pour cela, on peut construire un graphe appelé "NV-only", contenant uniquement les NV, avec une arrête entre deux NV si et seulement si ils partagent un NC en commun. Le nombre Γ d'emplacements mé-

moires est simplement donné par le nombre de couleurs dans ce graphe [132]. On peut donc appliquer un algorithme usuel de coloriage de graphes [133] pour déterminer Γ.

La discussion précédente montre aussi que le nombre Γ d'emplacements mémoires est déterminé par la structure du code LDPC considéré. C'est pourquoi, dans [128], nous avons proposé d'imposer une contrainte sur Γ directement lors de la construction du code LDPC. Ainsi, pour un protographe B donné, et pour une valeur de $\Gamma \geq d_{c,\max}$ fixée, nous avons proposé un algorithme PEG modifié qui construit le code LDPC de manière à satisfaire la condition sur Γ. Concrètement, l'algorithme PEG usuel [46] fonctionne en ajoutant une arrête l'une après l'autre dans le graphe. Habituellement, l'arrête à ajouter est choisie de manière à satisfaire une contrainte sur la girth (longueur du plus petit cycle) du code. Dans notre version de l'algorithme PEG, nous avons ajouté une condition supplémentaire, qui consiste à vérifier également que la condition sur Γ est satisfaite. Cela nécessite de construire, en plus du Tanner graph, le graphe VN-only décrit précédemment, et de déterminer si la l'ajout d'une arrête entre un NV et un NC permet toujours de satisfaire la condition sur le nombre de couleurs dans ce graphe.

5.3.5 Résultats numériques

Avec la construction précédente, il existe un compromis entre la girth du code et le nombre d'emplacements mémoires Γ que l'on doit utiliser. Par exemple, dans [128], nous avons considéré le protographe suivant :

$$B = \begin{bmatrix} 0 & 2 & 3 & 1 \\ 2 & 0 & 3 & 2 \end{bmatrix} \tag{5.20}$$

Dans ce protographe, le degré maximum est donné par $d_{c,\max} = 7$. Nous avons appliqué notre algorithme PEG modifié pour construire une matrice de base C avec un facteur d'extension $Z_1 = 36$. Nous avons observé qu'imposer $\Gamma = 7$ introduisait une nette dégradation de la performance du code LDPC, dûe au fait que dans ce cas, on obtenait un code de girth 4. Cependant, en relâchant un tout petit peu la contrainte en fixant $\Gamma = 8$, nous avons réussi à obtenir des codes ayant une performance quasiment identique au code construit sans contrainte sur Γ.

De plus, dans [128], nous avons considéré le décodage d'un code construit à partir du protographe B donné dans (5.20) avec $Z_1 = 36$ et $Z_2 = 18$, de dimension totale 1296×2592. Dans les deux cas, nous avons considéré un pipeline de profondeur $J = 20$.

Pour le premier code, nous avons observé qu'un nombre maximum de $L = 25$ itérations était nécessaire pour atteindre les mêmes performances en terme de BER qu'un décodeur utilisant un scheduling série sur les C-CN avec 20 itérations au maximum. Nous avons utilisé ce résultat pour montrer que notre architecture diminue la latence d'un facteur 3.2 comparé à une architecture utilisant le scheduling série.

Enfin, des résultats numériques liés à la synthèse de l'architecture réalisée avec un outil Cadence Genus pour une technologie 65nm sont disponibles dans [131] (espace occupé par l'architecture, estimation de la puissance, etc.). Ces résultats montrent qu'il existe un compromis entre la profondeur de pipeline et la consommation de puissance de l'architecture, qui est dû à une activité plus importante du circuit lorsque l'on utilise un pipeline plus profond.

5.4 Optimisation de la consommation d'énergie de l'architecture

Je présente maintenant les modèles de consommation d'énergie que nous avons proposé pour l'architecture précédente, ainsi que des méthodes pour optimiser certains paramètres du code, du décodeur, et de l'architecture, de manière à minimiser la consommation d'énergie. Je décris tout d'abord un modèle d'énergie pour un décodeur sans bruit, avant de présenter une extension au cas d'une architecture matérielle bruitée.

Dans toute cette partie, nous considérons un canal AWGN de variance σ^2 et de rapport signal-à-bruit (SNR) noté ξ.

5.4.1 Optimisation du protographe pour un modèle d'énergie seul

Dans [118], nous avons proposé un premier modèle d'énergie pour un décodeur non-bruité. Nous avons ensuite utilisé ce modèle pour optimiser le protographe du code LDPC. Ce modèle d'énergie nécessite une estimation du nombre moyen d'itérations nécessaires au décodage, et je présente donc tout d'abord une méthode pour estimer cette quantité.

Estimation du nombre moyen d'itérations

Pour estimer le nombre moyen d'itérations nécessaires au décodage d'un ensemble de codes représentés par un protographe B donné, nous avons utilisé la méthode d'évolution de densité à longueur finie de la manière suivante.

Dans la Section 3.4.2, nous avons vu que l'évolution de densité à longueur finie permettait de prédire la probabilité d'erreur du décodeur. Mais d'après [41], elle permet aussi de prédire le FER pour une longueur N donnée, en utilisant la formule suivante :

$$\mathrm{FER}_N^{(\ell)}(\xi) = \int_0^{\frac{1}{2}} \mathrm{FER}_\infty^{(\ell)}(z)\phi_{\mathcal{N}}\left(z; p, \frac{p(1-p)}{n}\right) dz. \tag{5.21}$$

Dans cette expression, le paramètre p est celui donné dans la Section 3.4.2 pour le canal AWGN, et $\mathrm{FER}_\infty^{(\ell)}(z) = 1-(1-P_e^{(\ell)}(z))^N$ représente le FER asymptotique, calculé à partir de la probabilité d'erreur $P_e^{(\ell)}$ évaluée par la méthode d'évolution de densité asymptotique.

A partir de l'expression précédente du FER, on peut prédire le nombre d'itérations moyen $L_N(\xi)$ à longueur N, en utilisant la formule suivante [118] :

$$L_N(\xi) = \sum_{\ell=1}^{L} \mathrm{FER}_N^{(\ell-1)}(\xi). \tag{5.22}$$

Cette expression est obtenue en supposant l'utilisation d'un critère d'arrêt "parfait", qui stoppe le décodeur si et seulement si le décodage est correct. Cependant, le critère d'arrêt classique qui consiste à vérifier les équations de parité du code peut stopper le décodage même si le résultat n'est pas correct (mauvais mot de code). En ce sens, l'expression précédente de $L_N(\xi)$ constitue une approximation mais qui a montré une bonne précision en simulations.

Modèles d'énergie

Dans [118], nous avons proposé deux modèles simples d'énergie : le premier évaluant la complexité du décodeur, et le second, le nombre d'accès mémoires nécessaires au décodage.

Le modèle de complexité évalué la consommation d'énergie E_c du décodeur en comptant le nombre d'opérations effectués dans l'architecture matérielle décrit dans la Sec-

tion 5.3. Nous avons obtenu l'expression suivante pour E_c [118] :

$$E_c(\xi) = L_N(\xi)N\Big(E_{\text{add}}(2(q+q_s)\tilde{d}_v + \frac{3}{2}(q-1)(\tilde{d}_c-1)+2)\Big) + L_N(\xi)N\Big(E_{\text{xor}}(2\tilde{d}_c+\tilde{d}_v-1)\Big). \tag{5.23}$$

Dans cette expression, $\tilde{d}_c$ et $\tilde{d}_v$ représentent les degrés moyens des NV et des NC du code, q représente le nombre de bits de quantification pour les messages échangés dans le décodeur, et $q+q_s$ représente le nombre de bits de quantification utilisés pour calculer les sommes totales $t_v^{(\ell)}$ des messages. De plus, les quantités E_{add} et E_{xor} représentent l'énergie nécessaire pour réaliser une addition et un XOR. Enfin, on voit que l'expression précédente dépend du SNR, à travers le nombre moyen d'itérations $L_N(\xi)$.

Le modèle de mémoire évalue la consommation d'énergie E_m du décodeur en comptant le nombre d'accès mémoires nécessaires au décodage. Nous avons obtenu l'expression suivante pour E_m [118] :

$$E_m(\xi) = L_N(\xi)NE_{\text{bit}}\Big((q+q_s)\tilde{d}_v + (1-R)(\tilde{d}_c+2(q-1))\Big) \tag{5.24}$$

Dans cette expression, la quantité E_{bit} représente l'énergie nécessaire à l'accès en lecture ou écriture d'un bit en mémoire.

Optimisation du protographe

Les modèles précédents dépendent d'un certains nombre de paramètres, et en particulier des degrés moyens $\tilde{d}_v$, $\tilde{d}_c$ des NV et des NC, et du nombre moyen d'itérations L_N nécessaires au décodage. Le protographe B du code LDPC aura une forte influence sur ces trois quantités, et c'est pourquoi nous proposons maintenant d'optimiser le protographe de manière à minimiser la consommation d'énergie du décodeur.

Dans [118], nous avons formulé le problème d'optimisation du protographe de la manière suivante :

$$\min_B E(\xi) \text{ tel que } \text{FER}_N^{(L)}(\xi) \leq \text{FER}_{\max}. \tag{5.25}$$

Dans cette expression, l'énergie E peut représenter soit E_c, soit E_m. Ce problème d'optimisation est donc formulé de manière à minimiser la consommation d'énergie, tout en respectant une contrainte de performance définie sur le FER.

Enfin, dans [118], nous avons proposé une méthode d'optimisation s'appuyant sur

TABLE 5.1 – Seuils et consommation d'énergie des protographes, optimisés pour la performance seule (B_0), pour le modèle de complexité (B_c), pour le modèle de mémoire (B_m), et pour les deux modèles (B_{opt}).

Protographe	Seuil	E_c	E_m
$B_{\text{opt}} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 0 & 4 & 1 \end{bmatrix}$	1.18 dB	35.9 nJ	508 nJ
$B_0 = \begin{bmatrix} 2 & 1 & 3 & 2 \\ 5 & 1 & 1 & 0 \end{bmatrix}$	1.15 dB	46.2 nJ	733 nJ
$B_c = \begin{bmatrix} 0 & 1 & 2 & 5 \\ 2 & 2 & 0 & 2 \end{bmatrix}$	1.20 dB	38.8 nJ	585 nJ
$B_m = \begin{bmatrix} 3 & 2 & 1 & 2 \\ 0 & 1 & 1 & 4 \end{bmatrix}$	1.21 dB	38.8 nJ	585 nJ

l'évolution différentielle [45] pour résoudre le problème d'optimisation précédent et trouver des protographes qui minimisent la consommation d'énergie du décodeur selon les modèles précédents.

Résultats numériques

Dans [118], nous avons considéré des paramètres particuliers $N = 10000$, $R = 1/2$, $q = 6$, $q_s = 2$, et un nombre maximum d'itérations $L = 50$. Nous avons appliqué la méthode d'optimisation précédente, pour optimiser des protographes de dimensions 2×4. Comme contrainte de performance, nous avons par exemple fixé $\text{FER}_{\max} = 10^{-2}$ pour un SNR cible $\xi = 1.45$dB. Nous avons obtenu quatre protographes : B_0, optimisé pour la performance uniquement (sans tenir compte de la consommation d'énergie), B_m, optimisé pour le modèle mémoire, B_c, optimisé pour le modèle de complexité, et enfin B_{opt}, optimisé vis à vis des deux modèles d'énergie. Les résultats d'optimisation sont présentés dans le Tableau 5.1, qui donne le seuil de chaque protographe et évalue sa consommation d'énergie selon chacun des modèles.

Les résultats d'optimisations sont bien ceux qui sont attendus : le protographe B_0 est celui qui a le meilleur seuil, au prix d'une consommation d'énergie plus élevée. De plus, le protographe B_{opt} minimise l'énergie à la fois pour le modèle de complexité, et pour le modèle de mémoire.

Enfin, dans cette partie, nous avons considéré des modèles d'énergie extrêmement simples, pas forcément réalistes. Mais cela nous a permis de développer une première approche, et surtout une méthode d'optimisation du protographe. Non seulement cette

méthode permet de prendre en compte plusieurs paramètres qui auront une influence sur la consommation d'énergie du décodeur (degrés moyens, nombre d'itérations, etc.), mais elle pourra aussi être adaptée facilement à des modèles d'énergie plus réalistes.

5.4.2 Optimisation de la consommation d'énergie en prenant en compte le bruit dans les mémoires

Pour terminer ce chapitre, nous présentons maintenant une extension du modèle de bruit précédent au cas d'un décodeur sujet à du bruit dans les mémoires. Dans [44], nous avons proposé un modèle d'énergie prenant cette fois en compte le bruit du circuit. Nous nous sommes concentrés sur le modèle d'énergie dans les mémoires, pour lequel il est plus simple de relier la consommation d'énergie au niveau de bruit. Nous présentons tout d'abord le modèle d'énergie que nous avons proposé, puis nous décrivons le problème d'optimisation que nous avons étudié.

Modèle d'énergie versus bruit

Dans cette partie, nous supposons que le bruit est introduit uniquement dans les mémoires, avec un taux d'erreur binaire ϵ. De plus, nous considérons que les mémoires sont alimentés par une tension V. Nous avons utilisé les résultats expérimentaux de [134] et [98] pour inférer le modèle suivant pour ϵ [44] :

$$\epsilon = \min\left(\exp\left(a + bV + cV^2\right), \frac{1}{2}\right), \tag{5.26}$$

Dans cette expression, les quantités a, b, c sont des paramètres du modèle, et dépendent de la technologie utilisée. Par exemple, pour une technologie récente 22nm, nous avons appliqué une régression polynomiale sur [134, Fig. 11] pour estimer $(a, b, c) = (22.12, -68.14, 0)$.

Ensuite, pour estimer la consommation d'énergie des mémoires, nous avons repris l'expression de E_m fournie dans (5.24), en écrivant simplement $E_{\text{bit}} = V^2/V_{\text{norm}}^2$. Cette expression correspond à l'énergie normalisée par bit mémoire, et V_{norm} est la tension d'alimentation nominale de la mémoire. Le nombre moyen d'itérations est toujours évalué à partir de l'évolution de densité à longueur finie (5.22), mais en considérant cette fois l'effet du bruit.

TABLE 5.2 – Valeurs d'énergie normalisées et paramètres optimaux pour différents protographes, avec différentes contraintes de performance.

Protographe	Cas 1 : $p_e^\star = 10^{-3}$, $\xi^* = 1.45$ dB					Cas 2 : $p_e^\star = 10^{-6}$, $\xi^* = 1.7$ dB				
	V_{opt}	q_{opt}	N_{opt}	E_{opt}	E_{nom}	V_{opt}	q_{opt}	N_{opt}	E_{opt}	E_{nom}
$B_{17} = [2\ 3\ 1\ 1\ ;\ 0\ 1\ 4\ 1]$	0.497	5	3060	342.4	845.2	0.532	7	6490	303.2	710.8
$B_{36} = [2\ 1\ 2\ 3\ ;\ 1\ 4\ 0\ 1]$	0.493	5	6170	377.13	931.1	0.539	6	7220	326.6	714.7
$B_m = [3\ 2\ 1\ 2\ ;\ 0\ 1\ 1\ 4]$	0.491	5	7700	347.73	869.9	0.533	5	7410	270.48	606.8
$B_c = [3\ 2\ 1\ 2\ ;\ 0\ 1\ 1\ 4]$	0.497	5	4070	365.93	890.2	0.533	6	5560	333.08	710.8

Optimisation de la consommation d'énergie

Ensuite, dans [44], nous avons considéré que le protographe était fixé, et proposé d'optimiser la consommation d'énergie du décodeur vis à vis de trois paramètres clés : le nombre de bits de quantification des messages q, la longueur du code N, la tension d'alimentation des mémoires V. Nous avons donc considéré le problème d'optimisation suivant :

$$\min_{V,q,N} E_m(V,q,N) \text{ tel que } P_{e,N}^{(L)}(V,q,N) \leq p_e^\star. \tag{5.27}$$

On cherche toujours à minimiser la consommation d'énergie du décodeur, mais cette fois, la contrainte de performance est exprimée en terme de probabilité d'erreur $p_e^\star$ estimé à partir de l'évolution de densité à longueur finie, voir l'équation (3.18). Notons que le critère de performance choisi n'a que peu d'impact sur la méthode d'optimisation en elle-même.

Dans [44], nous avons proposé une méthode d'optimisation pour résoudre le problème précédent. Au vu de la nature très différente des paramètres V, q, N, nous avons proposé une méthode d'optimisation alternée, consistant à résoudre successivement le problème (5.27) vis à vis d'un paramètre, en supposant que les deux autres sont fixés. Par exemple, on va tout d'abord fixer q et N, puis trouver le paramètre V qui minimise E_m tout en respectant la contrainte $P_{e,\max}$. Puis on va fixer q et V, et optimiser le paramètre N, et enfin faire de même avec q. L'optimisation alternée est ensuite répétée sur plusieurs itérations, jusqu'à convergence. Notons enfin que dans [44] nous avons proposé des heuristiques pour accélérer la recherche des paramètres optimaux à chaque itération.

Résultats numériques

Dans cette partie, nous considérons que le rendement du code est fixé à $R = 1/2$, et nous fixons différentes contraintes sur la performance, spécifiées dans le Tableau 5.2. On considère 4 protographes, B_{17} et B_{36} optimisés pour la performance seulement, ainsi que

B_m et B_c qui sont donnés dans le Tableau 5.1.

Le Tableau 5.2 fournit les résultats d'optimisation, et donne les paramètres optimaux obtenus V_{opt}, q_{opt} et N_{opt}, ainsi que le minimum d'énergie correspondant E_{opt} et l'énergie nominale E_{norm}. Après optimisation, on observe un gain conséquent en énergie, comparé au cas nominal. De plus, on observe que les paramètres optimaux varient en fonction des protographes et des contraintes de performance. En particulier, V_{opt} varie dans une plage de valeurs assez petite, tandis que N_{opt} varie beaucoup.

5.5 Conclusion

Dans ce chapitre, nous avons tout d'abord étudié l'effet du bruit sur les décodeurs LDPC, avec des modèles de bruit plus réalistes que dans les travaux existants. Puis nous avons proposé une architecture matérielle particulière, et étudié sa consommation d'énergie. En particulier, nous avons proposé un modèle simple reliant la consommation d'énergie des mémoires à l'effet du bruit dans les mémoires. Ces travaux nous on permis d'identifier les paramètres ayant un impact important sur la consommation d'énergie (protographe, longueur du code, etc.), et de mettre au point des méthodes d'optimisation efficaces. Dans la suite, il serait intéressant de considérer des modèles de consommation d'énergie en fonction du bruit plus réalistes.

Notons que dans [135], nous avons justement proposé un modèle d'énergie beaucoup plus réaliste, qui mesure l'activité du circuit. Les résultats de simulation montrent que ce modèle prédit avec beaucoup de précision la consommation d'énergie du circuit. Je ne l'ai pas présenté ici, car nous n'avons pas encore fait le travail d'extension au cas d'un circuit bruité. Mais cela serait un excellent point de départ pour des travaux futurs.

Chapitre 6

Effet du bruit dans les algorithmes de Machine Learning

6.1 Introduction

Dans la continuité du chapitre précédent, nous allons maintenant nous intéresser à d'autres méthodes de traitement de signal et de machine learning implémentées sur des circuits bruités. Tout comme les décodeurs LDPC, la plupart de ces méthodes sont conçues pour traiter des observations bruitées, et devraient donc être capables, dans une certaine mesure, de gérer également le bruit introduit par le circuit. De plus, dans certaines de ces méthodes, un résultat de calcul bruité aura un impact très limité. Par exemple, quand on réalise un test d'hypothèses, on compare une valeur réelle, souvent le log de la vraisemblance, à un seuil. Que cette valeur réelle contienne un bruit devrait faiblement impacter la décision finale, comme discuté dans [140].

6.1.1 Travaux existants sur le calcul sur circuit bruité

A la suite d'un travail pionnier de Taylor en 1968 [141], plusieurs travaux ont étudié l'effet du bruit dans des circuits logiques [142, 143, 144]. Ces travaux montrent qu'il est nécessaire d'ajouter des portes logiques supplémentaires dans le circuit pour garantir une probabilité d'erreur faible dans les sorties. Pour caractériser le nombre de portes logiques supplémentaires à ajouter, la notion de redondance a été définie dans [141] comme le nombre de portes logiques bruitées nécessaires à la réalisation d'un calcul fiable (avec une probabilité d'erreur fixée), divisée par le nombre de portes logiques non-bruitées nécessaires à la réalisation du même calcul. La redondance peut être vue comme une notion similaire à la capacité d'un canal de communications, mais uniquement des bornes inférieures et supérieures ont été fournies pour cette grandeur. En effet, l'effet du bruit dépend de la structure de calcul, et il en existe souvent une multitude pour réaliser la

même opération, ce qui rend difficile la caractérisation de la redondance optimale.

Plus récemment, [139, 138] se sont intéressés au calcul de fonctions logiques s'appuyant sur des portes XOR uniquement, et ont proposé des constructions s'appuyant sur des codes LDPC pour corriger les erreurs introduites par le circuit. De plus, suite aux résultats sur les tests d'hypothèses sur circuits bruités de [140], le problème de la régression logistique a été abordé dans [145]. Dans [140] comme dans [145], il n'est pas nécessaire d'ajouter de la redondance, en raison du peu d'impact du bruit dans ces méthodes simples de classification.

Enfin, des travaux plus pratiques sur l'implémentation matérielle de méthodes de traitement de signal se sont intéressés à l'effet de la quantification. Par exemple, le cas du filtrage de Kalman avec quantification des observations ou des états a été abordé dans [146, 147, 148]. Mais le bruit de quantification a souvent une nature différente du bruit du circuit, en particulier car il est à support borné. Si les travaux précédents se concentraient sur la caractérisation théorique de l'effet du bruit sur la qualité des méthodes étudiées, il existe aussi des études qui évaluent cet effet de manière empirique, par exemple pour des méthodes d'apprentissage tels que l'ACP ou des réseaux de neurones implementés dans des unités de calcul en mémoire [7, 149].

6.1.2 Problèmes étudiés

Dans ce chapitre, nous allons nous concentrer sur une classe de problèmes qui a été très peu étudié du point de vue de théorique : les algorithmes récursifs implémentés sur des circuits bruités. Dans ces algorithmes, que l'on peut exprimer sous la formulation générale $x_{k+1} = f(x_k)$, une question clé sera de déterminer dans quelle mesure un phénomène de propagation d'erreurs va affecter la qualité de la tâche de traitement. Pour étudier cette question, nous allons aborder trois types de problèmes :

1. **L'estimation binaire récursive**, qui permet d'estimer les états d'un modèle de Markov caché à états et observations binaires. L'étude de ce problème nous a permis de développer des premiers outils d'analyse de l'effet du bruit dans des opérations d'estimation récursives.
2. **Le filtrage de Kalman**, qui est un problème plus complexe d'estimation d'états à partir d'observations bruités. Nous avons analysé à la fois l'effet de la quantification et du bruit introduit dans les mémoires.
3. **Le calcul en mémoire**, qui permet de réaliser une partie des opérations de cal-

cul directement dans la mémoire, mais introduit un bruit non-négligeable dans les opérations de calcul. Nous avons étudié l'effet de ce bruit dans des multiplications matricielles, dans des méthodes de calcul de points fixes, et dans des réseaux de neurones.

6.2 Estimation binaire récursive

Dans cette section, nous présentons tout d'abord le problème de l'estimation binaire récursive dans sa version non-bruitée. Ensuite, nous introduisons notre modèle de bruit, puis l'analyse de l'effet du bruit dans la qualité de l'estimation.

6.2.1 Modèle de signal

On considère une suite d'états binaires $\{S_k\}_{k=1}^{+\infty}$ tels que $S_k \in \{0,1\}$ pour tout $k \geq 1$. Nous supposons que ces états sont distribués suivant un modèle de Markov tel que $P(S_k|S_{k-1},\cdots,S_1) = P(S_k|S_{k-1})$ pour tout $k \geq 1$. On note ensuite

$$\alpha = P(S_k = 1|S_{k-1} = 0) \quad \text{et} \quad \beta = P(S_k = 0|S_{k-1} = 1). \tag{6.1}$$

On dispose d'observations bruitées des états, notées X_k, avec une distribution de probabilité définie par

$$Q(i,j) = P(X_k = j|S_k = i), \quad i,j \in \{0,1\}. \tag{6.2}$$

Notons qu'il s'agit d'un cas particulier de modèles de Markov cachés [150].

Dans le problème d'estimation binaire récursive, on cherche à estimer la valeur de chaque état S_k à partir des observations $(X_1,\cdots,X_k)$. Dans le cas binaire, l'estimation peut être réalisée à partir d'une opération de filtrage, que nous décrivons maintenant.

6.2.2 Opération de filtrage non-bruitée

Pour estimer les états successifs, on souhaite réaliser une estimation au sens du maximum *a posteriori* de la forme

$$\hat{s}_k = \arg\max_{s\in\{0,1\}} P(S_k = s|x_1,\cdots,x_k). \tag{6.3}$$

Pour cela, on définit les rapports de vraisemblance

$$L_k = \log \frac{P(S_k = 1|x_1, \cdots, x_k)}{P(S_k = 0|x_1 \cdots, x_k)}.$$

D'après [151], on peut calculer les L_k successifs à partir de l'opération de filtrage suivante :

$$L_k = \log \frac{Q(1, x_k)}{Q(0, x_k)} + h(L_{k-1}) \tag{6.4}$$

où la fonction $h : \mathbb{R} \to \mathbb{R}$ a pour expression $h(y) = \log \frac{\alpha + e^y(1-\beta)}{(1-\alpha)+e^y\beta}$. On estime ensuite l'état S_k en fonction du signe de L_k.

6.2.3 Opération de filtrage bruitée

Dans [152], nous avons souhaité étudier l'effet du bruit et de la propagation éventuelle des erreurs dans la récursion (6.4) sur les L_k. Pour répondre à cette question, nous avons tout d'abord défini une version bruitée $\tilde{L}_k$ des rapports de vraisemblance, donnée par :

$$\tilde{L}_k = \log \frac{Q(1, x_k)}{Q(0, x_k)} + h(L_{k-1}) + B_k. \tag{6.5}$$

Dans cette expression, B_k est une variable aléatoire qui représente le bruit. La nature du bruit n'est pas spécifiée mais peut correspondre à un bruit ajouté lors du stockage de L_k dans la mémoire, ou à un bruit additif introduit lors du calcul de L_k.

De plus, nous avons souhaité faire aussi peu d'hypothèses que possible sur les variables B_k, avec l'objectif d'étudier une gamme de modèles de bruits aussi large que possible. Nous avons supposé que les B_k successifs étaient i.i.d, indépendants des S_k et des X_k. En plus de ces hypothèses usuelles, nous avons ajouté les conditions suivantes sur les moments d'ordre 1 des B_k :

$$\forall k \geq 1, E[B_k] = 0 \quad \text{et} \quad E[|B_k|] = \bar{B} < \infty. \tag{6.6}$$

En revanche, nous n'avons pas fait d'hypothèse suplémentaire sur les moments d'ordres supérieurs, ou sur la distribution de probabilité des B_k.

6.2.4 Étude de l'effet du bruit sur l'opération de filtrage

Nous montrons maintenant comment étudier l'effet du bruit B_k sur la qualité de l'estimation. Pour cela, nous allons étudier la quantité $E[|\tilde{L}_k - L_k|]$ qui exprime l'écart

moyen entre le filtrage avec bruit et le filtrage sans bruit. Dans [152], nous avons montré que cette quantité peut se borner de la manière suivante :

$$\overline{B} \leq E[|\tilde{L}_k - L_k|] \leq A_k. \tag{6.7}$$

De plus, la borne supérieure A_k a elle-même une expression récursive donnée par $= S(A_{k-1}) + \overline{B}$. La fonction S s'exprime comme :

$$S(b) = \log\left(1 + \frac{(e^b - 1)\mu}{2ve^{\frac{1}{2}b} + \alpha\beta e^b + (1-\alpha)(1-\beta)}\right) \tag{6.8}$$

avec $\mu = 1 - \alpha - \beta$, $v = \sqrt{\alpha\beta(1-\alpha)(1-\beta)}$. La fonction S est convexe, et pour tout $b \in \mathbb{R}$, on a $S(|b|) \geq 0$. Les bornes données dans (6.7) nous permettent de constater que le paramètre $\overline{B}$ a une grande influence sur l'écart entre la récursion bruitée et la récursion non-bruitée.

Ensuite, pour caractériser plus précisément la borne supérieure donnée dans (6.7), il est nécessaire d'étudier la convergence de la suite des A_k. Dans [152], nous avons montré que cette suite est croissante et positive, et qu'elle converge vers un unique point fixe $A^\star$ qui peut être calculé exactement en résolvant une équation du 4ème degré. De plus, nous avons obtenu la borne supérieure suivante sur $A^\star$:

$$\overline{B} \leq A^\star \leq \frac{\overline{B}}{1-\lambda}, \tag{6.9}$$

où $\lambda = \max_{x\in\mathbb{R}} S'(x)$.

Ce dernier résultat nous permet de conclure que l'opération de filtrage est robuste au bruit, dans le sens où il n'y a pas de propagation d'erreur. En effet, la récursion A_k converge, et la borne supérieure sur son point fixe $A^\star$ ne dépend que de la valeur de $\overline{B}$ et de la dérivée de la fonction S.

6.2.5 Résultats numériques

Nous présentons maintenant quelques résultats numériques liés à l'estimation binaire récursive bruitée. Nous fixons les valeurs de paramètres $\alpha = 0.05$, $\beta = 0.9$, $Q(0,1) = Q(1,0) = 0.1$, $N = 10000$ (la longueur de la chaîne de Markov). La Figure 6.1 représente la quantité $E[|\tilde{L}_k - L_k|]$ estimée par simulations numériques, ainsi que les bornes inférieures et supérieures fournies dans (6.7), et la borne supérieure sur le point fixe donné dans (6.9).

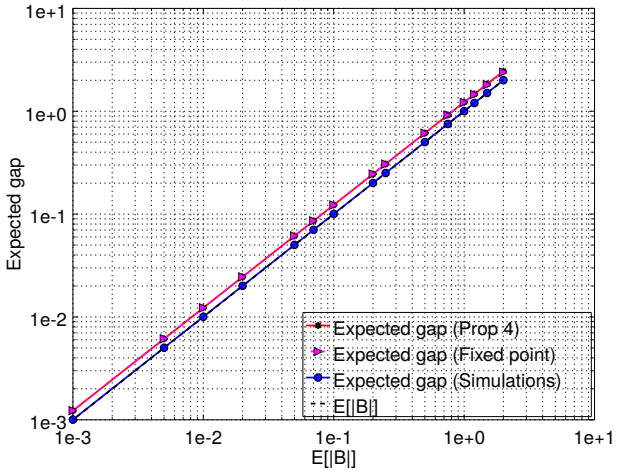

FIGURE 6.1 – Comparaison entre les bornes théoriques et les résultats de simulations pour l'estimation récursive bruitée. Les courbes rouge et violette sont superposées, et les courbes bleu et noire sont superposées également.

On constate tout d'abord que l'écart entre la borne supérieure et la borne inférieure est très faible, ce qui vient du fait que la valeur de λ dans (6.9) est elle-même très faible. On observe ensuite que les valeurs de bornes supérieures A_k sont superposées avec la borne sur le point fixe $\frac{\overline{B}}{1-\lambda}$. De manière plus surprenante, on observe que les résultats de simulations numériques sont superposées avec la borne inférieure $\overline{B}$. Dans [152], nous avons confirmé ces résultats pour plus de valeurs de paramètres α, β, $P(0,1)$ et $P(1,0)$, ce qui confirme la conclusion précédente sur la robustesse au bruit dans cet algorithme.

6.3 Filtre de Kalman

Par la suite, nous avons souhaité étudier des problèmes d'estimation plus complexes, et nous nous sommes donc concentrés sur le filtrage de Kalman. Comme dans la partie précédente, nous commençons par présenter la version non-bruitée du filtre, avant de décrire notre modèle d'erreur et notre méthode d'analyse de l'effet du bruit sur la qualité de l'estimation.

6.3.1 Modèle de signal

On considère une suite de vecteurs d'états $\mathbf{x}_k \in \mathbb{R}^c$ inconnus, et une suite de vecteur d'observations $\mathbf{y}_k \in \mathbb{R}^d$, avec le modèle linéaire suivant :

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{u}_k, \tag{6.10}$$

$$\mathbf{y}_{k+1} = H\mathbf{x}_k + \mathbf{v}_k. \tag{6.11}$$

Dans ces équations, la matrice F de dimension $c \times c$ est appelée matrice de transition, tandis que la matrice H de dimension $d \times c$ représente le modèle de mesure. Les vecteurs $\mathbf{u}_k$ et $\mathbf{v}_k$, de dimensions respectives c et d, représentent le bruit sur le modèle et le bruit sur les observations. Ils sont indépendants des $\mathbf{x}_k$, et possèdent des matrices de covariance connues P et Q.

6.3.2 Filtre de Kalman non-bruité

L'objectif du filtrage de Kalman est d'estimer les états successifs $\mathbf{x}_k$, à partir des observations $\mathbf{y}_k$ et de la connaissance du modèle précédent. Les expressions des estimées successives $\hat{\mathbf{x}}_k$ sont dérivées en cherchant à minimiser l'erreur quadratique moyenne $E[\|\mathbf{x}_k - \hat{\mathbf{x}}_k\|^2]$ à chaque instant k. Le filtre de Kalman se décompose en deux phrases. La première phase, appelée phase de prédiction, est définie par les équations suivantes :

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= F\hat{\mathbf{x}}_{k|k}, \\ P_{k+1|k} &= FP_{k|k}F^* + Q. \end{aligned} \tag{6.12}$$

La deuxième phase, appelée phase de correction, est décrite par les équations suivantes :

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + K_{k+1}(\mathbf{y}_{k+1} - H\hat{\mathbf{x}}_{k+1|k}), \\ K_{k+1} &= P_{k+1|k}H^*(HP_{k+1|k}H^* + R)^{-1}, \\ P_{k+1|k+1} &= (I - K_{k+1}H)P_{k+1|k}. \end{aligned} \tag{6.13}$$

Dans ces équations, les matrices de covariances successives $P_{k+1|k}$ ainsi que les matrices de gain K_{k+1} ne dépendent pas des observations et peuvent donc être calculées hors ligne.

Bien qu'étant une méthode assez ancienne, l'estimation par filtrage de Kalman est toujours utilisée dans de nombreuses applications actuelles, comme la détection de faux messages dans le protocole AIS [153], la prédiction de la qualité de l'air [154], ou le

positionnement de véhicules [155].

6.3.3 Modèle de bruit

Nous décrivons maintenant le modèle de bruit que nous avons considéré dans [156, 157] pour le filtrage de Kalman. Nous avons supposé que le bruit provenait de deux sources : la quantification, et le stockage de données en mémoire. En ce qui concerne la quantification, nous avons suivi l'approche de [107], et considéré un modèle standard de quantification à virgule fixe [158]. Dans ce modèle, chaque nombre réel est quantifié sur $(1 + n + m)$ bits et s'exprime comme

$$z = (-1)^{z_n} \sum_{b=-m}^{n-1} 2^b z_b, \tag{6.14}$$

où $\forall b \in [\![-m, n]\!]$, $z_b \in \{0, 1\}$ est une valeur binaire stockée en mémoire. En particulier, z_n correspond au bit de signe. Dans ce modèle, le paramètre n fixe la plus grande valeur qui peut représenter z, tandis que m donne la résolution de la quantification (plus petit écart entre deux valeurs quantifiées).

De plus, nous avons utilisé le modèle d'erreur dans la mémoire considéré dans [107]. Dans ce modèle, la valeur de chaque bit stocké en mémoire peut-être inversée avec une probabilité p. On peut donc représenter par une variable aléatoire Z_b la version bruitée de chaque bit z_b, et exprimer $Z_b = z_b + E_b$, où E_b prend ses valeurs dans $\{0, 1\}$ avec $P(E_b = 1) = p$. De plus, on suppose que le bit de signe z_n n'est pas affecté par le bruit, car cela pourrait causer des erreurs importantes. Cela revient à supposer que les cellules de mémoire dédiées aux bits de signe sont alimentées avec une puissance plus elevée que pour les autres bits.

A partir du modèle précédent défini sur les valeurs binaires, on peut définir un modèle d'erreur sur les valeurs quantifiées. On note Z la version bruitée de z donné dans (6.14), et on montre que $Z = z + B$, avec

$$B = (-1)^{z_n} \sum_{b=-m}^{n-1} 2^b E_b. \tag{6.15}$$

Si l'on suppose que les signes contenus dans z_n sont distribués de manière uniforme, on peut montrer que $E[B] = 0$ et $\mathrm{Var}[B] = (n + m + 1)2^{2b+1}p(1 - p)$.

Enfin, nous avons supposé que toutes les quantités impliquées dans le calcul du filtre sont quantifiées, pour rendre possible leur stockage en mémoire. Cela implique que non

seulement les vecteurs $\hat{\mathbf{x}}_{k+1|k}$, $\hat{\mathbf{x}}_{k+1|k+1}$, mais aussi les matrices $P_{k+1|k}$, $P_{k+1|k+1}$, K_{k+1}, sont quantifiées. En revanche, le bruit provenant du stockage en mémoire est introduit uniquement dans les estimées successives $\hat{\mathbf{x}}_{k+1|k}$ et $\hat{\mathbf{x}}_{k+1|k+1}$. En effet, on suppose que les autres quantités $P_{k+1|k}$, K_{k+1}, sont calculées hors-ligne et ne sont pas donc pas affectées par le bruit. Les vecteurs $\hat{\mathbf{x}}_{k+1|k}$ étant de dimension c, on considère que le bruit est ajouté composante par composante de manière indépendante et identiquement distribuée en suivant le modèle précédent. On note Γ la matrice de covariance de dimension $c \times c$ correspondante. La matrice Γ est diagonale et tous les termes diagonaux sont égaux à $\mathrm{Var}[B]$.

6.3.4 Effet du bruit dans le filtrage de Kalman

Dans [157], nous avons fourni des expressions analytiques des erreurs d'estimation *a priori* $\tilde{\mathbf{e}}_{k+1|k} = \tilde{\mathbf{x}}_{k+1|k} - \mathbf{x}_{k+1|k}$, et a *posteriori* $\tilde{\mathbf{e}}_{k+1|k+1} = \tilde{\mathbf{x}}_{k+1|k+1} - \mathbf{x}_{k+1|k+1}$. Dans ces expressions, $\tilde{\mathbf{x}}_{k+1|k}$ et $\tilde{\mathbf{x}}_{k+1|k+1}$ représentent les versions bruitées de $\hat{\mathbf{x}}_{k+1|k}$ et $\hat{\mathbf{x}}_{k+1|k+1}$. Cela nous a permis d'exprimer, toujours de manière analytique, les nouvelles matrices de covariances $P^*_{k+1|k}$ et $P^*_{k+1|k+1}$ de $\tilde{\mathbf{e}}_{k+1|k}$ et $\tilde{\mathbf{e}}_{k+1|k+1}$ sous les formes suivantes

$$P^*_{k+1|k} = F P^*_{k+1|k} F^T + Q + \Gamma + f(m) \tag{6.16}$$

$$P^*_{k+1|k+1} = (I - K_{k+1}H) P^*_{k+1|k} + \Gamma + g(m), \tag{6.17}$$

où $f(m)$ et $g(m)$ sont des fonctions avec des expressions relativement complexes, qui dépendent du nombre de bits de quantification et des paramètres du filtre [157].

Les expressions de ces matrices de covariances présentent plusieurs avantages. Tout d'abord, on peut remplacer le calcul de $P_{k+1|k}$ et $P_{k+1|k+1}$ dans les expressions du filtres de Kalman (6.12) et (6.13) par le calcul de $P^*_{k+1|k}$ et $P^*_{k+1|k+1}$, tout en gardant la même expression pour les autres termes. Cela aura pour effet de corriger les erreurs introduites par la quantification et le bruit, en plus du bruit d'observation. Deuxièmement, les expressions de $P^*_{k+1|k+1}$ permettent de prédire l'effet du bruit sur la performance du filtre de Kalman, puisque ses composantes diagonales représentent la variance de l'erreur d'estimation. Enfin, dans [156, 157], nous avons proposé d'utiliser les expressions de $P^*_{k+1|k+1}$ pour optimiser la consommation d'énergie du filtre, sous des contraintes de performance d'estimation.

6.4 Calcul en mémoire

Dans les études précédentes, nous avons fait l'hypothèse implicite d'une architecture matérielle de calcul standard, où les processeurs sont séparés physiquement des mémoires. Le problème de ce type d'architecture dit de Von Neumann, est qu'il nécessite des transferts de données conséquents entre mémoires et processeurs, ce qui est coûteux en énergie et augmente la latence. Cependant, dans le domaine du circuit, une alternative est en train d'émerger, qui consisterait à effectuer une partie des opérations de calcul directement dans la mémoire. En effet, les nouvelles technologies de mémoire non-volatile de type STT-MRAM [159], PCM [160], et ReRAM [103] permettent à la fois de stocker des valeurs dans leurs cellules mémoires, mais aussi d'effectuer des opérations de calcul analogiques, de type multiplication et accumulation [161]. Ces technologies émergentes s'appuient sur des composants résistifs particuliers avec des niveaux de résistance variables qui permettent de stocker l'information.

Cependant, ces technologies de mémoires sont particulièrement sensibles au bruit, à cause de plusieurs facteurs comme les variations liées à la fabrication [162], les processus d'écriture et de lecture dans les cellules mémoires [163], ou encore les effets de "sneak path" qui créent des chemins de courant indésirables dans les mémoires [164]. De plus, les mémoires de type PCM et ReRAM supportent le multi-niveau, dans le sens où il devient possible de stocker non pas deux valeurs différentes (qui correspondraient à un niveau 0 et à un niveau 1), mais un grand nombre de valeurs différentes dans chaque cellule mémoire [160, 103]. L'exploitation du multi-niveau semble particulièrement pertinente pour le calcul en mémoire, mais elle augmente la sensibilité au bruit puisque les marges entre les niveaux sont réduites [161].

Dans cette partie, je décris tout d'abord les architectures typiques de calcul en mémoire, et j'introduis un modèle de bruit particulier. Ensuite, je présente les méthodes que nous avons développées pour étudier l'influence du bruit sur le résultat des opérations de calcul, pour plusieurs applications comme la multiplication matricielle ou les réseaux de neurones.

6.4.1 Structure de calcul

Les unités de calcul en mémoire sont agencées sous la forme de crossbars en 2 dimensions, formées par l'intersection d'un certain nombre de "Word Lines" (WL) (les lignes) et de "Bit Lines" (BL) (les colonnes). Chaque intersection contient un composant résistif

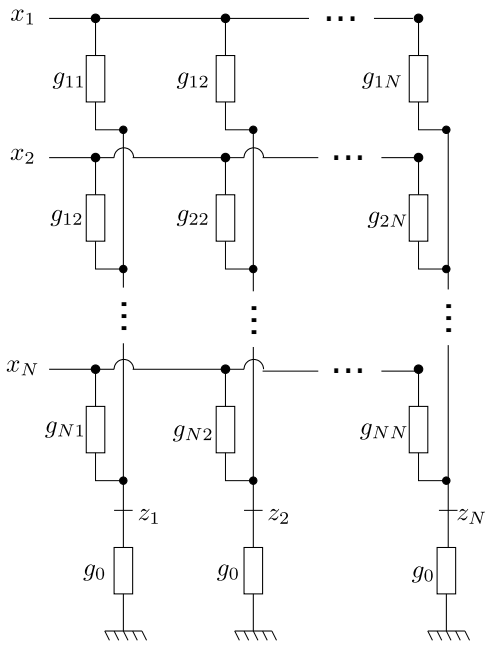

FIGURE 6.2 – Schéma d'une crossbar pour le calcul en mémoire

qui correspond à une cellule mémoire [7].

Un schéma typique de crossbar de dimension $N \times N$ est représenté en Figure 6.2. Sur ce schéma, les valeurs $g_{i,j}$ correspondent aux conductances internes des composants résistifs utilisés dans la mémoire. De plus, les valeurs $x_1, \cdots, x_N$ sont les tensions d'entrée, tandis que les valeurs $z_1, \cdots, z_N$ sont les tensions de sortie. Par application des relations classiques courant-tension (loi d'Ohm, loi des noeuds, etc.), on peut montrer que les tensions de sortie z_j ont pour expression [7]

$$z_j = \sum_{i=1}^{N} \frac{g_{i,j} x_i}{g_0 + \sum_{i'=1}^{N} g_{i',j}}. \tag{6.18}$$

En conséquence, un choix judicieux des valeurs de conductance $g_{i,j}$ et des tensions d'entrée $x_1, \cdots x_N$ permet d'effectuer une multiplication $\mathbf{z} = A\mathbf{x}$ entre une matrice A et un vecteur $\mathbf{x} = [x_1, \cdots, x_N]^T$.

Ceci dit, l'approche précédente pour le calcul en mémoire souffre de plusieurs limitations [7, 161]. Premièrement, les dimensions des crossbars sont assez limitées, et typiquement il est difficile d'aller au delà de valeurs de N supérieures à 1024. En conséquence, pour des tailles de matrice A plus importantes, plusieurs crossbar devront être utilisées. Ensuite, les valeurs de conductance peuvent uniquement être positives. Si la matrice A

contient des valeurs négatives, la solution usuelle est d'utiliser deux crossbars, une pour les valeurs positives et une pour les valeurs négatives, et de combiner les sorties des deux crossbars pour obtenir le résultat final. De plus, en pratique, les valeurs de $g_{i,j}$ sont quantifiées sur un certain nombre de niveaux.

Dans la suite de cette section, nous allons supposer pour simplifier qu'un calcul matriciel est réalisé à partir d'une seule crossbar qui peut-être de taille arbitraire, et que les $g_{i,j}$ peuvent prendre n'importe quelle valeur réelle. Je discuterai dans une dernière partie des extensions que nous avons proposé pour éliminer ces hypothèses.

6.4.2 Modèle de bruit

En plus des limitations précédentes, il est raisonnable de considérer que les valeurs de $g_{i,j}$ sont bruitées [162, 163, 7, 161], ce qui risque d'affecter le résultat de l'opération (6.18). Pour modéliser ce bruit, nous avons proposé de représenter les vraies valeurs de conductances comme des variables aléatoires $G_{i,j}$ admettant des moments d'ordre 1 et 2 [165]. De plus, nous avons supposé que $\mathbb{E}[G_{i,j}] = g_{i,j}$, où $g_{i,j}$ représente la valeur cible de conductance, et nous avons noté $\mathbb{V}\mathrm{ar}[G_{i,j}] = \sigma^2$. Nous n'avons en revanche effectué aucune hypothèse supplémentaire sur la forme de la densité de probabilité des variables aléatoires $G_{i,j}$.

Ceci dit, nous avons supposé que les $G_{i,j}$ sont i.i.d., ce qui n'est pas forcément le cas en pratique. En particulier, [166] montre que la variance d'une conductance dépend de sa valeur cible. De plus, il peut exister une dépendance statistique entre les $G_{i,j}$, en fonction de leurs positions respectives dans la crossbar [167]. Mais l'hypothèse i.i.d. permet de simplifier l'analyse dans un premier temps, et a été considérée dans les quelques autres travaux théoriques existants sur ce sujet, voir [168] par exemple.

Dans nos travaux, nous avons utilisé ce modèle particulier pour caractériser de manière analytique l'effet du bruit dans les opérations de calcul, pour plusieurs application : la multiplication matricielle seule et son extension au calcul de points fixes [165], ainsi que différentes architectures de réseaux de neurones [169, 170].

6.4.3 Multiplication matricielle

Dans [165], nous avons étudié le problème de la multiplication matricielle selon la structure de calcul décrite dans la Section 6.4.1. Nous avons supposé que les tensions d'entrées sont elles aussi des variables aléatoires X_i, indépendantes, et telles que $\mathbb{E}[X_i] = x_i$

et $\mathbb{V}\text{ar}(X_i) = \gamma_i^2$. Nous avons ensuite étudié la convergence en distribution des variables aléatoires Z_j correspondantes aux tensions de sortie. En appliquant le théorème central limite et le théorème de Slutsky, nous avons montré le résultat suivant :

$$\frac{N^2}{\sqrt{v_j}}(Z_j - z_j) \overset{d}{\Rightarrow} \mathcal{N}\left(0, \frac{1}{\alpha_j^2}\right) \tag{6.19}$$

où $\overset{d}{\Rightarrow}$ représente la convergence en distribution. Dans cette expression, $\alpha_j = \lim_{N\to\infty} \frac{1}{N^2} \sum_{i=1}^{N} g_{i,j}$ est une limite finie et différente de 0, tandis que v_j est un terme d'espérance donné dans [165]. Le résultat précédent permet d'affirmer que quelque soient les distributions des $G_{i,j}$ et des X_i, les sorties Z_j sont approximativement Gaussiennes, centrées sur la valeur cible z_j et de variance $\frac{v_j}{N^4\alpha_j^2} \to 0$ quand $N \to \infty$.

Ce résultat permet de montrer que les sorties aléatoires Z_j ont un bon comportement, puisqu'elles sont centrées sur les valeurs cibles z_j. De plus, de manière intéressante, quand le nombre d'entrées X_i augmente, la variance des Z_j tend vers 0, ce qui signifie que le bruit se compense et que l'on retrouve la bonne valeur de z_j à la sortie. Cependant, il s'agit d'un résultat de convergence, et donc valable pour N assez grand. De plus, ce résultat suppose que les tensions d'entrées X_i sont indépendantes statistiquement. Mais en pratique, les X_i ne sont pas nécessairement indépendantes, en particulier parce que ces tensions ont elle-mêmes été calculées à partir d'autres crossbars. Et les sorties d'une crossbar, par exemple les Z_j ici, ne sont pas indépendantes, car elles sont calculées à partir des mêmes entrées. C'est pourquoi, dans la suite, nous proposons une autre analyse qui permet de caractériser l'effet du bruit lorsque l'on considère plusieurs multiplications matricielles successives.

6.4.4 Multiplications matricielles successives

Dans de nombreuses applications, il est nécessaire d'appliquer plusieurs multiplications matricielles successives de manière à calculer

$$\mathbf{y} = A^{(T)}A^{(T-1)} \cdots A^{(1)}\mathbf{x}, \tag{6.20}$$

où les matrices $A^{(t)}$ sont toutes de dimension $N \times N$. Dans [7], il est montré que différents problèmes d'optimisation (optimisation avec et sans contraintes, analyse en composante principale, etc.) peuvent être résolus à partir d'une opération de la forme de (6.20). En particulier, si $A^{(t)} = A$ pour tout $t \in [\![1, T]\!]$, cette opération récursive permet de calculer les points fixes de A.

Dans notre configuration, le calcul de (6.20) revient à utiliser T crossbars, chacune correspondant à une des matrices $A^{(t)}$. Pour cela, on note $\mathbf{x}^{(t)} = A^{(t)}\mathbf{x}^{(t-1)}$, et sa version bruitée $\mathbf{X}^{(t)} = A^{(t)}\mathbf{X}^{(t-1)}$. Dans ce cas de figure, on ne peut pas appliquer les résultats de la Section 6.4.3 sur la distribution de probabilités des sorties, car pour $t \leq 1$, les composantes de $\mathbf{X}^{(t)}$ ne sont pas indépendantes, ce qui empêche d'utiliser le théorème central limite.

Méthode d'évolution des moments

C'est pourquoi nous avons développé une deuxième méthode d'analyse théorique, qui consiste à calculer les moments d'ordre 1 et 2 de manière itérative pour chaque $\mathbf{X}^{(t)}$. En particulier, on définit $\mu_j^{(t)} = \mathbb{E}[X_j^{(t)}]$, $\rho_j^{2(t)} = \mathbb{V}\mathrm{ar}[X_j^{(t)}]$, $\rho_{j,j'}^{(t)} = \mathbb{C}\mathrm{ov}(X_j^{(t)}, X_{j'}^{(t)})$, comme les espérances, variances, covariances, des composantes $X_j^{(t)}$ des $\mathbf{X}^{(t)}$, pour tout $j \in [\![1, N]\!]$, et pour tout $t \in [\![1, T]\!]$. Les termes d'espérance et de variance serviront ensuite à exprimer l'erreur quadratique moyenne, sous la forme

$$\mathrm{MSE}(t) = \frac{1}{N}\sum_{j=1}^{N}(\rho_j^{2(t)} + (\mu_j^{(t)} - x_j^{(t)})^2). \tag{6.21}$$

Les termes de corrélation $\rho_{j,j'}^{(t)}$ seront eux utiles pour exprimer les $\mu_j^{(t)}$ et $\rho_j^{2(t)}$ successifs.

Expressions des moments

Pour exprimer les moments précédents, on note $T_j = \sum_{i=1}^{N} G_{ij}^{(t)} X_i^{(t-1)}$ et $\Delta_j = G_0^{(t)} + \sum_{i=1}^{N} G_{ij}^{(t)}$. Ensuite, dans [165], nous avons montré que les moments peuvent s'exprimer de la manière suivante :

$$\mu_j^{(t)} = \frac{\mathbb{E}[T_j]}{\mathbb{E}[\Delta_j]} - \frac{\mathbb{C}[T_j, \Delta_j]}{\mathbb{E}[\Delta_j]^2} + \frac{\mathbb{V}[\Delta_j]\mathbb{E}[T_j]}{\mathbb{E}[\Delta_j]^3} \tag{6.22}$$

$$\rho_j^{2(t)} = \frac{\mathbb{V}[T_j]}{\mathbb{E}[\Delta_j]^2} + \frac{3\mathbb{E}[T_j]^2\mathbb{V}[\Delta_j]}{\mathbb{E}[\Delta_j]^4} - \frac{4\mathbb{E}[T_j]\mathbb{C}[\Delta, T_j]}{\mathbb{E}[\Delta_j]^3} \tag{6.23}$$

$$\rho_{j,j'}^{(t)} = \sum_{i=1}^{N}\sum_{i'=1}^{N} \mathbb{E}\left[\frac{G_{i,j}^{(t)}}{\Delta_j}\right] \mathbb{E}\left[\frac{G_{i',j'}^{(t)}}{\Delta_{j'}}\right] \rho_{i,i'}^{(t-1)} \tag{6.24}$$

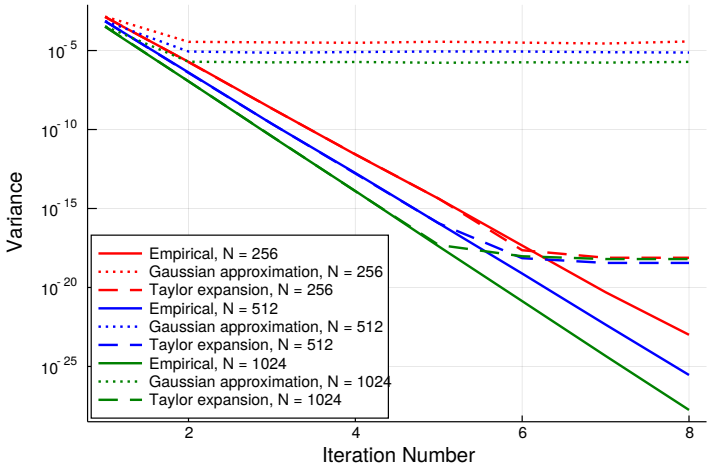

FIGURE 6.3 – Variances théoriques et empiriques en fonction de l'itération t, pour plusieurs dimensions N de crossbars.

avec $\mathbb{E}[T_j] = \sum_{i=1}^{N} g_{ij}^{(t)} \mu_i^{(t-1)}$, $\mathbb{E}[\Delta_j] = g_0 + \sum_{i=1}^{N} g_{ij}^{(t)}$, $\mathbb{V}[\Delta_j] = N\sigma^2$, $\mathbb{C}[\Delta_j, T_j] = \sigma^2 \sum_{i=1}^{N} \mu_i^{(t-1)}$, et $\mathbb{C}[G_{i,j}^{(+)}, \Delta_j^{(+)}] = \sigma^2$. De plus, on a

$$\mathbb{V}[T_j] = \sigma^2 \sum_{i=1}^{N} (\rho_i^{2(t-1)} + \mu_i^2) + \sum_{i=1}^{N} \sum_{i'=1}^{N} g_{i,j}^{(t)} g_{i',j}^{(t)} \rho_{i,i'}^{(t-1)}, \tag{6.25}$$

$$\mathbb{E}\left[\frac{G_{i,j}^{(t)}}{\Delta_j}\right] = \frac{\mathbb{E}[G_{i,j}^{(t)}]}{\mathbb{E}[\Delta_j]} - \frac{\mathbb{C}[G_{i,j}^{(t)}, \Delta_j]}{\mathbb{E}[\Delta_j]^2} + \frac{\mathbb{V}(\Delta_j)\mathbb{E}[G_{i,j}^{(t)}]}{\mathbb{E}[\Delta_j]^3} \tag{6.26}$$

On obtient donc des expressions des moments à l'instant t en fonction des moments à l'instant $t-1$, en tenant bien compte des corrélations introduites dans les sorties bruitées. La difficulté principale dans la dérivation de ces équations a résidé dans l'expression de moments d'ordre 1 et 2 de rapports de variables aléatoires de la forme $\frac{T_j}{\Delta_j}$. En effet, il n'existe pas d'expression exacte de ces moments, sauf pour des distributions particulières [171]. C'est pourquoi, nous avons utilisé des développements de Taylor à l'ordre 2 [172], qui nous ont permis d'obtenir les expressions précédentes. Celles-ci ne sont pas exactes, mais fournissent de très bonnes approximations, comme nous le verrons dans les résultats numériques.

Résultats numériques

Je présente maintenant quelques résultats numériques liés au développement théorique précédent. On considère $T = 8$ multiplications matricielles successives, selon l'opération

décrite par l'équation (6.20), et on génère aléatoirement 8 matrices successives $A^{(t)}$. Les bruits sur les conductances sont également générés aléatoirement, suivant des lois Gaussiennes de variance $\sigma_{i,j}^2$ (aussi fixée aléatoirement) pour chaque $g_{i,j}$. Ensuite, pour plusieurs dimensions de crossbars, $N = 256$, $N = 512$, $N = 1024$, on calcule la variance théorique fournie par l'équation (6.22), que l'on compare à la variance estimée à partir de simulations de Monte Carlo. La Figure 6.3 représente les variances théoriques et empiriques en fonction de l'itération t, ainsi que la variance obtenue à partir de l'approximation Gaussienne (6.19), c'est à dire en négligeant les corrélations à l'intérieur des vecteurs successifs $\mathbf{x}^{(t)}$.

On observe tout d'abord que jusqu'à $t = 5$, les variances théoriques et empiriques sont parfaitement confondues, quelque soit la longueur N considérée. Ce premier résultat confirme la précision du développement théorique proposé dans cette partie, et cela malgré le développement de Taylor à l'ordre 2 qui ne fournit qu'une approximation de la variance. Ceci dit, à partir de $t = 5$, on observe une saturation des courbes théoriques, qui s'écartent par la même occasion des courbes empiriques. Cela vient probablement d'erreurs de précision numérique, car les quantités mises en jeu sont extrêmement faibles (la variance est autour de 10^{-18}). On observe aussi que l'approximation Gaussienne n'est pas suffisante pour prédire les courbes empiriques, et qu'il est donc essentiel de prendre en compte les corrélations dans les vecteurs $\mathbf{x}^{(t)}$.

Enfin, on note que la variance décroît rapidement au cours des itérations successives. Cela vient probablement du fait que les bruits dans les crossbars se compensent.

6.4.5 Réseaux de Neurones

Les réseaux de neurones constituent une application importante du calcul en mémoire, tant ils sont devenus standards dans beaucoup d'applications. Dans cette partie, nous nous concentrons sur les réseaux de type Multi-Layer Perceptron (MLP) standards, et nous discuterons rapidement d'extensions par la suite.

Description du réseau

On s'intéresse maintenant à un réseau de neurones constitué de T couches successives. Dans ce cas, on peut décrire complètement le réseau à partir des relations suivantes, pour

$t \in [\![1, T]\!]$:

$$\mathbf{x}^{(t)} = A^{(t)}\mathbf{y}^{(t-1)} \tag{6.27}$$

$$\mathbf{y}^{(t)} = \mathcal{S}(\mathbf{x}^{(t)}), \tag{6.28}$$

où la fonction $\mathcal{S} : \mathbb{R} \to \mathbb{R}$ est appliquée sur chaque composante de $\mathbf{y}^{(t)}$ et correspond à la fonction d'activation (sigmoïde, Relu, etc.) du réseau de neurones. On suppose que la partie linéaire (6.27) de la couche est réalisée à partir d'une crossbar bruitée, tandis que la partie non-linéaire (6.28) est réalisée dans un circuit CMOS standard.

Méthode d'évolution des moments

On souhaite maintenant mettre à jour l'évolution des moments présentée dans la section 6.4.4, pour prendre en compte les non-linéarités introduites par les fonctions d'activation. Pour cela, on considère les versions bruitées $\mathbf{X}^{(t)}$ et $\mathbf{Y}^{(t)}$ de $\mathbf{x}^{(t)}$ et $\mathbf{y}^{(t)}$. On définit comme avant $\mu_j^{(t)} = \mathbb{E}[X_j^{(t)}]$, $\rho_j^{2(t)} = \mathbb{V}\mathrm{ar}[X_j^{(t)}]$, $\rho_{j,j'}^{(t)} = \mathbb{C}\mathrm{ov}(X_j^{(t)}, X_{j'}^{(t)})$. On a maintenant aussi besoin d'exprimer les moments d'ordre 1 et 2 des $\mathbf{y}^{(t)}$, à savoir $\nu_i^{(t)} = \mathbb{E}[Y_i^{(t)}]$, $\gamma_i^{2(t)} = \mathbb{V}\mathrm{ar}[Y_i^{(t)}]$, $\gamma_{i,i'}^{(t)} = \mathbb{C}\mathrm{ov}(Y_i^{(t)}, Y_{i'}^{(t)})$.

On souhaite maintenant exprimer les relations récursives entre les moments successifs. D'après [169], les expressions de $\mu_j^{(t)}$, $\rho_j^{2(t)}$, $\rho_{j,j'}^{(t)}$ données en Section 6.4.4 sont toujours valables, à condition de remplacer les $\mu_i^{(t-1)}$, $\rho_i^{2(t-1)}$, $\rho_{i,i'}^{(t-1)}$ par $\nu_i^{(t-1)}$, $\gamma_i^{2(t-1)}$, $\gamma_{i,i'}^{(t-1)}$, respectivement. De plus, en utilisant à nouveau des développements de Taylor à l'ordre 2, on montre que

$$\nu_i^{(t)} = \mathcal{S}\left(\mu_i^{(t)}\right) + \frac{1}{2}\mathcal{S}''\left(\mu_i^{(t)}\right)\rho_i^{2(t)} \tag{6.29}$$

$$\gamma_i^{2(t)} = \frac{1}{2}\mathcal{V}''(\mu_i^{(t)})\rho_i^{2(t)} - f(\mu_i^{(t)})\mathcal{S}''(\mu_i^{(t)})\rho_i^{2(t)} \tag{6.30}$$

$$\gamma_{i,i'}^{(t)} = \mathcal{S}'(\mu_i^{(t)})\mathcal{S}'(\mu_{i'}^{(t)})\rho_{i,i'}^{(t)} \tag{6.31}$$

avec $\mathcal{V} = \mathcal{S}^2$. L'application de l'expression de l'erreur quadratique moyenne donnée dans (6.21) permet ensuite d'étudier l'effet du bruit dans les couches successives du réseau de neurones.

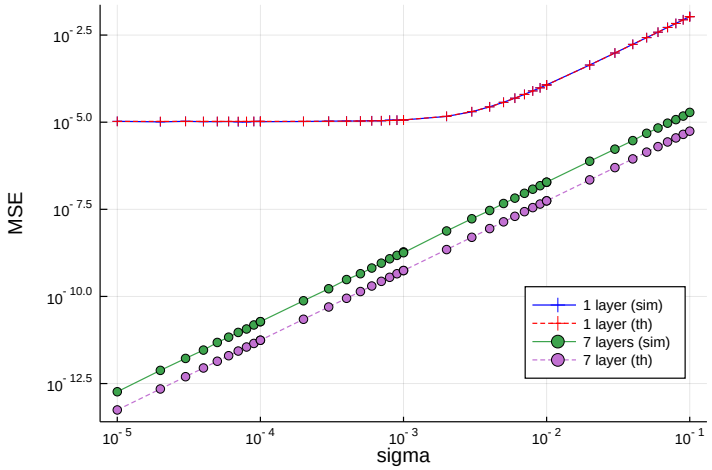

FIGURE 6.4 – Erreur quadratique moyenne (MSE) en fonction de σ, pour un réseau de neurones à 7 couches, après la première couche et après la dernière couche.

6.4.6 Résultats numériques

Je présente maintenant quelques résultats numériques pour le cas des réseaux de neurones. On considère un réseau de neurones à 7 couches, de dimensions respectives $(100, 100, 200, 150, 120, 80, 10)$, et des poids aléatoires sur chaque couche. On utilise la fonction sigmoïde comme fonction d'activation. La Figure 6.4 représente l'erreur quadratique moyenne en fonction de σ, à la fois calculée à partir des formules théoriques précédentes, et mesurée à partir de simulations de Monte Carlo. Les erreurs sont calculées après la première couche du réseau, et après la septième couche du réseau.

Après la première couche, on observe que les courbes théoriques et empiriques sont parfaitement superposées. Après la dernière couche, les deux courbes ont des comportements très proches, même si on observe un petit écart enter les deux courbes. Cela vient probablement des développements de Taylor successifs nécessaires au calcul théorique des moments. Ces résultats confirment malgré tout la précision et l'intérêt de l'analyse théorique proposée dans ce chapitre. Pour optimiser un réseau en terme de performance ou d'énergie, il sera en effet beaucoup plus efficace en terme de temps de calcul de s'appuyer sur les résultats de l'analyse théorique.

6.4.7 Extensions

Nous avons étendu les travaux précédents à différents cas de figures intéressants en pratique. Tout d'abord, les équations précédentes supposaient implicitement que tous les

coefficients des matrices $A^{(t)}$ étaient positifs, ce qui est bien sûr loin d'être le cas en pratique. Pour étudier le cas général où les coefficients peuvent être positifs ou négatifs, nous avons utilisé l'architecture décrite dans [7] qui consiste à utiliser deux crossbars par calcul matriciel, une pour les coefficients positifs, et une pour les coefficients négatifs. Dans [169] nous avons étendu notre méthode d'évolution des moments à ce cas de figure.

De plus, dans [170], nous avons considéré une architecture de calcul en mémoire légèrement différente du schéma de la Figure 6.2, qui consiste à remplacer la résistance finale de conductance g_0 par un amplificateur à transimpédance. Cela permet d'obtenir la relation

$$z_j = r_s \sum_{j=1}^{n} g_{i,j} x_i, \tag{6.32}$$

où r_s est une valeur de résistance, à la place de (6.18). Dans ce cas, on peut directement mapper les coefficients d'une matrice A dans les valeurs de conductance, à savoir $g_{i,j} = a_{i,j}$. Cependant, cette simplicité apparente se fait au prix d'une augmentation de la consommation d'énergie de la crossbar, due aux amplificateurs. Dans [170], nous avons étendu la méthode d'évolution des moments à cette deuxième architecture, qui est plus simple à analyser que la première. En particulier, dans ce cas, on peut exprimer les moments d'ordre 1 et 2 après la couche linéaire de manière exacte, sans avoir à utiliser les développements de Taylor. De plus, nous avons aussi fourni des expressions analytiques des moments après la couche non-linéaire, sous forme d'intégrales, que nous avons proposé d'évaluer numériquement. Cela a permis d'améliorer la précision de l'évaluation des moments. Enfin, dans [170], nous avons considéré des valeurs quantifiées de conductance, ce qui correspond davantage à un cas pratique.

Enfin, dans des travaux soumis récemment, nous avons étendu l'analyse à d'autres réseaux de neurones de type CNN. Ce type de réseau est plus difficile à analyser en raison des couches convolutives qui introduisent des corrélations supplémentaires.

6.5 Conclusion

Dans ce chapitre, nous avons utilisé des outils de probabilités et statistiques (calcul de distributions de probabilités et de moments) pour caractériser l'effet du bruit du circuit dans diverses méthodes de traitement de signal et de machine learning, ayant en commun d'utiliser des algorithmes récursifs. Nous avons considéré à la fois des architectures standards de calcul de type Von Neumann (séparation des unités de calcul et des mémoires),

et des technologies émergentes de calcul en mémoire. Nos résultats numériques pour les différentes méthodes montrent que les analyses théoriques que nous avons développé prédisent avec une très bonne précision les résultats des simulations de Monte Carlo. De plus, les résultats numériques confirment que la plupart des méthodes sont relativement robustes, bien sûr jusqu'à un certain niveau de bruit.

Dans chacun des travaux présentés dans ce chapitre, en plus d'effectuer l'analyse théorique de l'effet du bruit, nous avons considéré des modèles simples reliant le niveau de bruit à la consommation d'énergie du système. Cela nous a permis de proposer des méthodes d'optimisation de l'énergie, de manière à satisfaire une certaine performance cible. Les résultats numériques d'optimisation que nous avons obtenu nous on permis de montrer les gains importants en énergie, dûs également à la bonne robustesse au bruit des méthodes étudiées. Ces méthodes d'optimisation n'ont pas été présentées dans ce chapitre, mais elles sont décrites en détails dans les différents articles que nous avons publié sur ces sujets.

Enfin, une perspective importante de ces travaux consisterait à réfléchir à l'utilisation de codes correcteurs d'erreurs pour rendre ces méthodes encore d'avantages robustes au bruit, lorsque celui-ci est introduit à un niveau élevé.

Chapitre 7

Travaux en cours et perspectives

7.1 Introduction

Dans ce dernier chapitre, je décris mes travaux en cours et perspectives, qui se concentrent sur trois sujets : 1) le calcul en mémoire, 2) l'apprentissage sur données codées, 3) le stockage de données dans l'ADN. Ces trois sujets présentent des connections avec le domaine des codes correcteurs d'erreurs, comme évoqué en introduction.

7.2 Calcul en mémoire

J'ai décrit mes travaux sur le calcul en mémoire dans le chapitre précédent, dans la Section 6.4. Je présente maintenant mes perspectives sur ce sujet.

7.2.1 Étude de différentes applications

Dans le chapitre précédent, nous avons supposé que les unités de calcul en mémoire étaient utilisées uniquement pour réaliser des opérations de multiplication matricielle analogique, qui sont utilisés dans beaucoup d'algorithmes (méthodes d'optimisation, ACP, etc.) [7], incluant les réseaux de neurones. Mais les unités de calcul en mémoire peuvent être utilisés différemment, par exemple pour évaluer des distances de Hamming [168], implémenter des circuits logique en mémoire [173, 174, 175, 176], ou calculer des plus courts chemins [177]. Il serait intéressant d'évaluer l'effet du bruit dans ces applications également, ce qui a été fait uniquement de manière empirique dans les travaux existants. Dans le même esprit, il serait intéressant d'étudier l'effet du bruit dans des architectures de calcul en mémoire conçues pour l'entraînement de réseaux de neurones [178].

Dans ces différentes applications, un point clé est de déterminer le bon critère pour caractériser l'effet du bruit. Dans nos études précédentes, nous avons principalement consi-

déré l'erreur quadratique moyenne, qui a du sens pour le calcul matriciel ou si l'on s'intéresse à des problèmes de régression. En revanche, pour des problèmes de classification, l'accuracy serait un critère plus adapté. Pour du calcul de distance de Hamming ou de la logique en mémoire, on souhaiterait étudier des probabilités d'erreur. La difficulté dans ce cas réside dans l'évaluation statistique de ces critères en fonction des caractéristiques du bruit.

7.2.2 Modèles de bruit plus réalistes

Dans les études précédentes, nous avons principalement modélisé les erreurs comme du bruit additif au niveau des composants résistifs, et nous avons supposé une indépendance statistique entre les bruits dans la crossbar. Une première extension évidente consisterait à intégrer à l'analyse des modèles de bruit plus complexes, incluant notamment des dépendances statistiques en fonction des positions des cellules mémoires dans la crossbar [167].

Il serait intéressant également d'étudier d'autres types d'erreurs, comme les erreurs de switching entre états possibles des cellules mémoires (par exemple, dans une mémoire à deux niveaux possibles, un niveau haut qui deviendrait un niveau bas), ou des fautes de type "stuck-at", où des cellules mémoire défaillantes sont bloquées à un niveau donné. Enfin, les effets de type sneak-path mériteraient d'être étudiés également.

7.2.3 Un équivalent de la capacité pour le calcul en mémoire

Dans le contexte des communications numériques, la notion de capacité permet de prédire les débits atteignables par un système de codage sur un canal bruité. Dans le cas du calcul bruité, des tentatives ont était faites pour définir et étudier une notion équivalente. En particulier, dans le cas de circuits logiques bruités, la redondance est définie comme le nombre de portes logiques nécessaires au calcul bruité divisé par le nombre de portes logiques pour le circuit non-bruitée, de manière à atteindre une certaine probabilité d'erreur [142, 143, 141, 144]. Il existe uniquement des résultats d'atteignabilité pour la redondance d'un circuit logique bruité, qui est en effet une notion difficile à étudier car l'effet du bruit dépend fortement du type de circuit considéré. Il serait cependant intéressant de généraliser cette définition de la redondance à des structures de calcul en mémoire pour des applications plus larges que les circuits logiques.

Ceci dit, la notion de redondance ne prend pas en compte certains autres effets qui pourraient améliorer la robustesse au bruit. Par exemple, en augmentant les valeurs re-

latives des conductances des cellules mémoires, on augmente mécaniquement le rapport signal à bruit et donc la robustesse. Pour avoir une compréhension plus fine de ces différents effets, il serait intéressant de pouvoir caractériser la puissance nécessaire à la crossbar pour effectuer un calcul avec un bon niveau de fiabilité. Cela permettrait de capturer à la fois les effets de la redondance (ajout de codes correcteurs d'erreurs) et de l'amplitude des valeurs de conductance. Cela permettrait aussi d'étudier d'autres phénomènes, comme le coût des convertisseurs analogiques/numériques entre les circuits de type CMOS et les crossbars.

7.2.4 Conception de codes correcteurs d'erreurs

Enfin, une étape fondamentale et plus pratique consisterait à concevoir des codes correcteurs d'erreurs efficaces permettant de fiabiliser le calcul en mémoire. Les solutions standards de codes correcteurs d'erreurs ne peuvent pas être utilisées ici, car elles supposent une linéarité dans le corps binaire GF(2), alors que l'opération de calcul en mémoire consiste en une somme pondérée analogique dans $\mathbb{R}$. Quelques solutions ont été proposées [179], qui permettent de corriger un nombre maximum d'erreurs, ou qui s'appuient sur des constructions extrêmement complexes [162]. En conséquence, la construction de codes correcteurs d'erreurs efficaces, peu complexes, et adaptés à l'application (multiplication matricielle seule, réseaux de neurones, logique en mémoire, etc.), reste encore un problème largement ouvert.

7.3 Codage source/canal pour l'apprentissage

Dans beaucoup d'applications où il est nécessaire de transmettre des données, l'objectif n'est pas de reconstruire l'information originale, mais plutôt d'effectuer une tâche d'apprentissage sur les données : prise de décisions, classification, recommandation de contenu, etc. Le sujet récent des goal-oriented communications [180] propose donc de concevoir le système de codage source/canal des données de manière à considérer directement l'objectif d'apprentissage, à la place des critères habituels de reconstruction (probabilité d'erreur, distortion, etc.). C'est un sujet qui a pris beaucoup d'ampleur récemment, à l'interface entre le domaine des télécommunications et de l'apprentissage automatique. Il existe maintenant de nombreux travaux récents en théorie de l'information et en codage de sources, bien qu'un certain nombre de questions restent encore largement ouvertes.

7.3.1 Travaux existants

Dans cette partie, je présente tout d'abord les travaux existants sur les aspects analyse de théorie de l'information et construction de schémas pratiques, avant de décrire mes perspectives sur ces sujets dans la section suivante.

Analyse de théorie de l'information

Du point de vue du codage source/canal, une première question clé réside dans la détermination des performances limites au sens de la théorie de l'information pour différents problèmes d'apprentissage sur données codées. La difficulté réside dans la modélisation du problème (que sont les sources, quelles sont les variables à apprendre, etc.), et dans la prise en compte de critères non-usuels dans l'analyse de la théorie de l'information.

Plusieurs travaux ont étudié ces sujets. Tout d'abord, [181] montre que le débit nécessaire à l'estimation de paramètres dans un schéma de codage avec information adjacente est inférieur au débit nécessaire à la reconstruction des données, tandis que [182] obtient des bornes inférieures et supérieures sur l'erreur de généralisation d'un problème d'apprentissage. Les bornes fournies dans [182] s'appliquent à une large variété de problèmes d'apprentissage, mais dans [183], nous avons montré que la borne supérieure était très mauvaise dans le cas d'un problème simple de régression. Ensuite, dans [18], le problème des goal-oriented communications est formalisé comme l'idée de reconstruire deux sources X et Z, seul Z étant observé à l'encodeur, avec deux contraintes différentes de distortion sur chacune des sources. Enfin, des approches de type information-bottleneck [184] ont été proposées pour caractériser les performances des schémas d'apprentissage sur données codées, en utilisant des critères d'information mutuelle. Cependant la difficulté majeure de ces travaux reste de transformer un problème d'apprentissage formalisé par exemple en terme de performance de classification, en un problème de minimisation de distortion ou d'information mutuelle.

Dans un autre registre, il existe beaucoup de travaux sur les tests d'hypothèses distribués, ou l'objectif du décodeur est de prendre une décision H_0 ou H_1 à partir d'observations compressées des sources [24, 185, 186, 187]. Ces travaux caractérisent cette fois une quantité propre à un problème de test d'hypothèse : l'exposant d'erreur de Type-II. Ils étudient différents types de tests (test contre l'indépendance, etc.) et différentes configurations de communications (avec et sans canal, schéma à accès multiple, etc.).

Une analyse de théorie de l'information devrait aussi permettre d'étudier une autre

question fondamentale, à savoir le compromis entre le critère de reconstruction des données et le critère d'apprentissage. Il est essentiel d'étudier ce compromis pour de nombreuses applications où les données sont déjà compressées, et où l'on souhaite appliquer une tâche d'apprentissage sans avoir à décompresser une quantité gigantesque de données. Dans ce type de configuration, plusieurs travaux ont identifié un compromis entre la reconstruction et l'apprentissage en terme de débit de codage. Par exemple [21] montre qu'il existe un compromis entre un critère de distortion sur les données, et un critère de divergence lié à la perception. De même, [188] considère un problème de test d'hypothèse distribué, et identifie un compromis entre la reconstruction des données et l'erreur de Type-II du test. Enfin [189] étudie le compromis entre reconstruction et identification dans une base de données bruitée. Le seul exemple notable d'absence de compromis provient de [183], où nous avons montré que pour un problème de régression, le même schéma de codage est optimal pour les deux objectifs.

Construction de schémas pratiques

Du côté de la construction de schémas pratiques, plusieurs travaux ont proposé d'estimer les paramètres [190], de réaliser des tests d'hypothèse [191] ou du clustering [192, 193], à partir d'un petit nombre de combinaisons linéaires des données d'entrée, en suivant une approche de type Compressed Sensing (CS)[194]. Cependant, ces travaux ne sont pas directement adaptés à la transmission de données numériques, car ils produisent des données réelles et n'évaluent pas l'effet de la quantification ou du bruit du canal sur les performances d'apprentissage. Dans une tentative de développer des approches discrètes de CS pour l'apprentissage, dans [72] nous avons proposé des méthodes d'estimation de paramètres à partir de données codées avec des codes LDPC, et dans [195] nous avons étudié le clustering de données à travers des codes LDPC. Mais ces travaux considèrent uniquement des données binaires, ce qui limite leur portée pratique.

Dans le domaine du codage source/canal, des approches plus systématiques et adaptées à des vrais données (images par exemple) ont été proposées. Certaines familles de solutions considèrent des méthodes de compression standard, comme JPEG, et proposent d'utiliser des réseaux de neurones au décodeur pour effectuer une classification, après avoir reconstruit uniquement les coefficients transformés [196], ou même sans effectuer de reconstruction préalable [197]. Dans [198], nous avons conservé l'idée du réseau de neurones au décodeur, mais proposé un encodage un peu différent et plus adapté à la tâche d'apprentissage à réaliser au décodeur. Ces deux dernières approches peuvent être qualifiées

d'hybrides, avec des décodeurs basés Deep-Learning, et des encodeurs présentant malgré tout des structures standard, avec une quantification, une transformée, etc. Il existe aussi des approches Deep-Learning de bout en bout pour des critères d'apprentissage, en particulier à base d'auto-encoders [199, 200, 201, 202]. Mais ces schémas doivent être entraînés hors-ligne au préalable, et nécessitent donc une connaissance assez fine des modèles de source et canal sous-jacents, alors que les solutions précédentes peuvent être entraînées en ligne, car leur encodeur est fixe.

7.3.2 Perspectives

Je présente maintenant mes perspectives sur ce sujet.

Performances de théorie de l'information à longueur finie

Des problèmes relativement simples, comme le test d'hypothèse [24, 185, 186, 187] ou la régression [183] ont été bien étudiés du point de vue de la théorie de l'information. La simplicité de ces problèmes réside dans leurs critères de performance : probabilités d'erreur de type I et II pour les tests d'hypothèses, erreur quadratique moyenne pour la régression. En réalité, ces critères sont assez similaires aux critères classiques de probabilité d'erreur et de distortion pour la reconstruction des données. Il semble cependant essentiel de considérer des tâches d'apprentissages plus complexes et largement utilisées en pratique, comme la classification. Cela nécessite d'adresser certaines questions clé, comme le critère de performance à considérer, qui devra être à la fois raisonnable du point de vue de l'apprentissage, et possible à incorporer dans une analyse de théorie de l'information. De plus, cela nécessite une réflexion approfondie sur les schémas d'atteignabilité pour les preuves de théorie de l'information.

De plus, un point important selon moi est que les problèmes d'apprentissage sont des problèmes de codes courts. En effet, il suffit en principe de quelques dizaines de bits pour prendre une décision correcte dans le cas d'un test d'hypothèse, et il ne devrait pas y avoir besoin de beaucoup plus pour de la classification. Il est donc essentiel de proposer des analyses non-asymptotiques, et de s'appuyer sur des outils de théorie de l'information à longueur finie, comme ceux développés par Kostina dans [203].

Etude du compromis reconstruction/apprentissage

Ensuite, une extension importante du point de vue de la théorie de l'information réside dans l'étude du compromis entre la tâche de reconstruction et la tâche d'apprentissage. Là aussi, il serait souhaitable d'étudier davantages de problèmes d'apprentissage et surtout de critères de performance, les études actuelles se limitant principalement à des critères de distortion [18], ou de divergence [21]. Il serait intéressant d'étudier dans quelle mesure le compromis dépend de la tâche d'apprentissage en elle-même, ou alors davantage du critère d'évaluation considéré.

Schémas pratiques

Enfin, une perspective essentielle selon moi réside dans la conception de schémas de codage pratiques pour les communications goal-oriented. Si des solutions pratiques existent dans le domaine de l'apprentissage automatique sur données compressées [190, 191, 192, 193], elles négligent les contraintes pratiques (quantification, etc.) associées à la transmission de l'information. Peu de solutions issues du domaine du codage ont été proposées, et cela constitue donc un manque important. Même pour des problèmes simples comme les tests d'hypothèses ou la régression, il n'existe pas de solution pratique efficace s'appuyant sur des codes source ou canal.

Un point qui me semble important serait de considérer des approches hybrides décrites précédemment, avec un encodeur sous un format classique (quantification, transformée, etc.), quitte à adapter ces étapes à l'objectif final, et un décodeur utilisant éventuellement un réseau de neurones. Cela permettrait de considérer plus facilement les différents objectifs (reconstruction et apprentissage) ensemble, et aussi d'adapter la partie apprentissage en ligne, en fonction des données reçues et de la tâche à effectuer.

7.4 Stockage de données dans des molécules d'ADN

Ces dernières années, le stockage de données dans des molécules d'ADN synthétiques a suscité beaucoup d'intérêt, et émergé comme une alternative possible aux supports de stockage traditionnels (disques durs, bandes magnétiques, etc.). En effet, il a une densité et une durée de vie bien supérieures aux supports classiques, et peut se conserver à température ambiante [204]. Ce nouveau support présente aussi de nombreux défis, non seulement du point de vue des techniques de bio-informatique à mettre en oeuvre, mais

aussi pour la protection et l'intégrité des données. Dans cette partie, après avoir présenté le principe général du stockage de données dans l'ADN, je décris certaines perspectives du point de vue de la théorie de l'information et du codage.

7.4.1 Principe

Une molécule d'ADN peut se représenter comme une succession de bases, ou nucléotides, de quatre types : A,C,G, et T. Pour pouvoir stocker de l'information, on utilise une technique de synthèse qui permet de transformer une séquence numérique dans un alphabet quaternaire en une molécule d'ADN particulière [205]. Il existe différentes techniques de synthèse, comme la synthèse chimique, qui est notamment utilisée pour fabriquer des médicaments avec des principes actifs particuliers. Ce processus extrêmement coûteux (environ 1000 euros pour synthétiser une séquence) et générateur de déchets, représente aujourd'hui l'un des goulots d'étranglements du stockage de données dans l'ADN [206]. L'opération de synthèse produit plusieurs copies de la même molécule d'ADN, qui sont ensuite amplifiées au moyen d'une technique de PCR [205].

Dans un second temps, lorsque l'on souhaite lire l'information contenue dans la molécule d'ADN, on réalise un séquençage, une technique qui a permis de décoder le génome humain et celui de multiples autres organismes vivants, et qui a connu un développement important lors des dernières décennies. Cette technique est beaucoup moins onéreuse que la synthèse, puisqu'il il est possible d'acquérir des séquenceurs "de bureau" pour quelques milliers d'euros. Parmi les séquenceurs usuels, on peut citer Illumina et Oxford Nanopore. Dans cette dernière solution, la molécule d'ADN passe à travers un nanopore, qui renvoie une série de courants électriques correspondant à des blocs de 6 bases successives, appelées 6-mers [207]. On utilise ensuite un outil appelé basecaller, qui va transformer la série de valeurs de courant en une suite de bases, ce qui permet de retrouver la séquence d'origine.

7.4.2 Défis et opportunités du point de vue du codage

Autant la synthèse est un processus extrêmement fiable qui ne produit pas d'erreur dans la molécule d'ADN, autant l'opération de séquençage introduit une proportion importante d'erreurs, entre 5 et 10%, dans la séquence numérique en sortie du basecaller [204, 208]. De plus, ces erreurs ne sont pas de celles que l'on rencontre habituellement dans le domaine des télécommunications. En effet, il peut s'agir d'insertions, c'est à dire que des symboles non désirés sont insérés à des positions aléatoires et inconnues. Il peut aussi

s'agir de déletions, où des symboles sont purement et simplement supprimés à des positions aléatoires et inconnues. Il existe enfin des substitutions, où des symboles sont remplacés aléatoirement par d'autre symbole, qui sont des erreurs plus classiques. Un enjeu important dans ce domaine consiste donc à développer des solutions de correction efficaces, permettant de corriger les trois types d'erreur. Il n'est évidement pas possible d'utiliser directement les solutions de codes classiques LDPC, Turbo, Polaires, etc. car leurs décodeurs standards sont incapables de corriger les insertions et déletions [209, 210, 211, 212].

Cependant, il existe aussi une opportunité intéressante pour le codage canal : l'opération de séquençage produit en réalité un grand nombre de lectures de la même séquence d'entrée. Cela est dû au fait que la synthèse et l'amplication par PCR produisent un grand nombre de copies de la même molécules d'ADN [213, 214]. De plus, les réalisations des erreurs varient d'une lecture à l'autre. Il est donc essentiel d'exploiter ces lectures multiples lors de la conception du système de correction des erreurs, pour augmenter son efficacité.

Dans la suite, je présente brièvement mes premiers travaux sur le sujet du stockage de données dans l'ADN, avant d'identifier des perspectives à plus long terme.

7.4.3 Contributions

Jusqu'à maintenant, nous avons proposé deux types de contributions : un modèle de canal pour le stockage de données dans l'ADN, et des solutions de correction d'erreurs adaptées à ce modèle.

Modèle de canal

Dans un premier temps, nous avons souhaité proposer un modèle statistique de canal représentant le processus de stockage de données dans l'ADN, et prenant en compte la synthèse, le stockage en lui-même, le séquençage, et le basecaller. Alors que beaucoup de travaux existants considèrent que les erreurs sont indépendantes et identiquement distribuées (i.i.d.) [210, 212, 213], nous avons souhaité prendre en compte la dépendance des erreurs à la séquence d'entrée, ainsi que la mémoire dans les évènements d'erreurs successifs. Ces deux phénomènes, que nous avions observé dans les données expérimentales, s'expliquent par la lecture en 6-mers lors du séquençage par nanopore. Concrètement, on note $\mathbf{x} = (x_1, \cdots, x_N)$ la séquence d'entrée de longueur N, et $(e_1, \cdots e_N)$ la séquence de longueur N

d'évènements produits par le canal, avec $\forall k \in [\![1, N]\!]$, $e_k \in \{\text{INS}, \text{DEL}, \text{SUB}, \text{MATCH}\}$. Puis on décrit le modèle de canal par un ensemble de probabilités [208, 215]

$$P(e_k|e_{k-1}, x_{k-5}, \cdots, x_k) \tag{7.1}$$

prenant en compte l'évènement précédent e_{k-1} et le 6-mer courant $(x_{k-5}, \cdots, x_k)$. Les valeurs de ces probabilités sont ensuite estimées à partir de données expérimentales. Nos résultats publiés dans [208, 215] montrent que notre modèle présente une fidélité accrue vis à vis des données, mesurée avec la divergence de Kulback-Leibler, par rapport aux modèles existants [216, 217, 218].

Obtenir un modèle de canal réaliste permet de tester extensivement de nouvelles méthodes de codage "in-silico", avant de mettre en oeuvre des expériences "in-vivo" complexes et coûteuses. De plus, il est essentiel d'exploiter la connaissance des statistiques du canal dans la solution de correction d'erreurs, pour améliorer sa performance.

Solutions de codage canal

Nous avons proposé deux solutions de codage distinctes, présentant des caractéristiques différentes.

La première solution a été pensée pour optimiser au maximum la performance de décodage. Nous nous sommes appuyés sur une solution de codage proposée dans [213], qui est construire à partir d'un code concaténé, constitué d'un code LDPC externe et d'un code convolutif interne. Le code convolutif est décodé à l'aide d'un algorithme BCJR modifié qui permet de corriger les trois types d'erreurs, au prix d'une complexité élevée dûe à l'introduction de nouveaux états dans le trellis. Ensuite, on utilise un décodeur LDPC standard pour corriger les erreurs de substitution résiduelles, ce qui permet de diminuer le plancher d'erreur de manière significative. Dans [208], nous avons repris la même construction concaténé, et proposé un nouvel algorithme BCJR qui prend en compte notre modèle de canal à mémoire. Nous avons ajouté des états supplémentaires dans le trellis, pour prendre en compte la dépendance aux évènements précédents et à la séquence d'entrée. Nos résultats de simulation montrent que la solution proposée est très efficace pour corriger les erreurs du canal de stockage ADN, permettant d'atteindre un taux d'erreur par trame (FER) d'environ 10^{-3} en exploitant seulement 2 ou 3 lectures.

Mais outre la complexité très importante du décodage, l'inconvénient majeur des solutions proposées dans [208, 213] réside dans le faible rendement de codage, en général

inférieur à 1/2. Cela représente un coût très important en terme de synthèse, puisque pour un rendement 1/4 utilisé dans [208, 213], on aura 1 bit d'information pour 4 bits stockés. Nous avons donc développé une deuxième solution de correction d'erreurs [219], avec l'objectif d'utiliser des rendements de codage élevés. Pour cela, nous utilisons uniquement un code LDPC de rendement supérieur à 1/2. Après le séquençage, nous utilisons un algorithme de consensus initialement proposé dans le domaine de la bio-informatique, qui permet de reconstruire une unique séquence à partir d'un grand nombre de lectures, mais sans exploiter le code. Ensuite, nous avons proposé un algorithme de synchronisation, qui utilise la structure du code LDPC pour corriger les insertions, deletions, et substitutions résiduelles après le consensus. Les résultats de [219] montrent que pour un code LDPC de rendement 3/4, il est possible d'atteindre un FER de 10^{-3} en utilisant cette fois une trentaine de lectures. Ce dernier nombre peut paraître important, mais il ne faut pas oublier que la synthèse coûte très cher, tandis que le séquençage produit de toute manière un grand nombre de lectures des séquences.

7.4.4 Perspectives

Je présente maintenant mes perspectives sur le sujet du stockage de données dans l'ADN.

Calcul de la capacité

Plusieurs travaux récents se sont attachés à estimer la capacité du canal de stockage de données dans l'ADN [220, 221, 222]. Les difficultés principales ont consisté notamment à prendre en compte les erreurs de type insertions et déletions lors du calcul de la capacité, ainsi que les lectures multiples. Ces travaux considèrent des modèles peu réalistes, et il pourrait être intéressant d'étudier également des modèles plus pratiques, comme celui que nous avons proposé dans [215].

Conception de codes à haut rendement

La première perspective qui me semble importante à court terme, réside dans la conception de codes de haut rendements plus efficaces. S'il est en effet essentiel d'utiliser des codes de haut rendement pour réduire les coûts de synthèse, il existe encore un écart important entre les deux solutions décrites précédemment, en terme de nombre de lectures qu'il est nécessaire d'utiliser pour reconstruire l'information. Pour réduire cet écart, une

solution serait de remplacer l'algorithme de consensus utilisé dans [219] par une solution plus efficace, qui prenne par exemple en compte la connaissance de notre modèle de canal.

Intégration de la solution de codage canal dans un schéma de stockage complet

De manière générale, il n'est pas suffisant de penser la solution de codage canal toute seule, sans réfléchir aux interactions avec les autres étapes du stockage de données dans l'ADN. Pour citer un premier exemple, l'opération de synthèse introduit des contraintes particulières au niveau de la composition des séquences que l'on peut synthétiser : il faut en particulier éviter les longues répétitions de symboles, appelées homopolymers, et respecter un équilibre entre les bases de type A,C, et celles de type G,C (il faut environ 50% de chaque). Pour respecter ces contraintes, de nombreuses solutions ont été proposées [223, 224, 225], qui diminuent un peu le rendement du code. Il semble essentiel de réfléchir à l'interaction entre ces codes à contraintes et les codes correcteurs d'erreurs. Ces codes sont en effet appliqués juste avant la synthèse et modifient donc la séquence codée. De plus, leur décodage suppose en général qu'il n'y a pas d'erreur dans la séquence. Il faut donc réfléchir à comment intégrer ces codes à contraintes lors de la correction d'erreur.

Un autre élément important réside dans le fait que pour gagner en densité, la tendance est qu'une solution d'ADN stockée ne contienne pas un unique type de molécule représentant une seule séquence, mais un grand nombre de molécules différentes correspondant à des séquences distinctes. Une solution consiste à appliquer une méthode de clustering [226] pour séparer les lectures correspondant à des séquences distinctes. Il semble donc important de réfléchir à l'impact de cette étape sur la correction des erreurs, tout d'abord pour réduire l'impact des erreurs de clustering (qui pourraient être vues comme des séquences outliers). De plus, étant donné que beaucoup d'algorithmes de clustering ont pour effet dérivé de construire des "représentants" (ou centroïdes) des différents clusters, il pourrait être intéressant de voir comment exploiter ces représentants pour effectuer une première passe de correction d'erreurs.

Calcul en mémoire

Enfin, des travaux très récents dans les domaines de la bio-chimie et de la bio-informatique ont montré qu'il était aussi possible d'effectuer du calcul en utilisant des molécules d'ADN [227, 228]. Selon ces travaux qui sont pour le moment de l'ordre de la preuve de concept, il serait possible de synthétiser des opérations et circuits logiques

très simples à partir de molécules d'ADN. Cela permettrait de réaliser des circuits de calcul dans l'ADN, massivement parallèles et nécessitant extrêmement peu d'énergie. Si les technologies bio-chimiques doivent encore largement progresser sur ce sujet, une question intéressante reste de se demander, comme pour le cas du stockage, comment corriger les erreurs de séquençage, qui risquent de corrompre les résultats des calculs.

BIBLIOGRAPHIE

[1] Claude Elwood SHANNON, « A mathematical theory of communication », in : *The Bell system technical journal* 27.*3* (1948), p. 379-423.

[2] Claude BERROU, Alain GLAVIEUX et Punya THITIMAJSHIMA, « Near Shannon limit error-correcting coding and decoding : Turbo-codes. 1 », in : *IEEE International Conference on Communications (ICC)*, t. 2, IEEE, 1993, p. 1064-1070.

[3] Robert GALLAGER, « Low-density parity-check codes », in : *IRE Transactions on Information Theory* 8.*1* (1962), p. 21-28.

[4] David JC MACKAY, Simon T WILSON et Matthew C DAVEY, « Comparison of constructions of irregular Gallager codes », in : *IEEE Transactions on Communications* 47.*10* (1999), p. 1449-1454.

[5] Erdal ARIKAN, « Channel polarization : A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels », in : *IEEE Transactions on Information Theory* 55.*7* (2009), p. 3051-3073, DOI : 10.1109/TIT.2009.2021379.

[6] Zixiang XIONG, Angelos D LIVERIS et Samuel CHENG, « Distributed source coding for sensor networks », in : *IEEE Signal Processing Magazine* 21.*5* (2004), p. 80-94.

[7] Sijia LIU et al., « A memristor-based optimization framework for artificial intelligence applications », in : *IEEE Circuits and Systems Magazine* 18.*1* (2018), p. 29-44.

[8] Masayuki TANIMOTO, « FTV (free-viewpoint television) », in : *APSIPA Transactions on Signal and Information Processing* 1 (2012), e4.

[9] Gene CHEUNG, Antonio ORTEGA et Ngai-Man CHEUNG, « Interactive streaming of stored multiview video using redundant frame structures », in : *IEEE Transactions on Image Processing* 20.*3* (2010), p. 744-761.

[10] Shinya SHIMIZU et al., « View scalable multiview video coding using 3-d warping with depth map », in : *IEEE Transactions on Circuits and Systems for Video Technology* 17.*11* (2007), p. 1485-1495.

[11] Gordon E Moore, « No exponential is forever : but" Forever" can be delayed ![semiconductor industry] », in : *IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. IEEE, 2003, p. 20-23.

[12] Puneet Gupta et al., « Underdesigned and opportunistic computing in presence of hardware variability », in : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32.*1* (2012), p. 8-23.

[13] Bing Li, Bonan Yan et Hai Li, « An overview of in-memory processing with emerging non-volatile memory for data-intensive applications », in : *Great Lakes Symposium on VLSI*, 2019, p. 381-386.

[14] Miao Hu et al., « Dot-product engine for neuromorphic computing : Programming 1T1M crossbar to accelerate matrix-vector multiplication », in : *Annual Design Automation Conference*, 2016, p. 1-6.

[15] Christopher N Takahashi et al., « Demonstration of end-to-end automation of DNA data storage », in : *Scientific reports* 9.*1* (2019), p. 4998.

[16] Reinhard Heckel, Gediminas Mikutis et Robert N Grass, « A characterization of the DNA data storage channel », in : *Scientific reports* 9.*1* (2019), p. 9663.

[17] Changsheng Gao et al., « Towards task-generic image compression : A study of semantics-oriented metrics », in : *IEEE Transactions on Multimedia* (2021).

[18] Photios A Stavrou et Marios Kountouris, « A rate distortion approach to goal-oriented communication », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2022, p. 590-595.

[19] Emilio Calvanese Strinati et Sergio Barbarossa, « 6G networks : Beyond Shannon towards semantic and goal-oriented communications », in : *Computer Networks* 190 (2021), p. 107930.

[20] Lingyu Duan et al., « Video coding for machines : A paradigm of collaborative compression and intelligent analytics », in : *IEEE Transactions on Image Processing* 29 (2020), p. 8680-8695.

[21] Yochai Blau et Tomer Michaeli, « The perception-distortion tradeoff », in : *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, p. 6228-6237.

[22] David Slepian et Jack Wolf, « Noiseless coding of correlated information sources », in : *IEEE Transactions on Information Theory* 19.*4* (1973), p. 471-480.

[23] Aaron Wyner et Jacob Ziv, « The rate-distortion function for source coding with side information at the decoder », in : *IEEE Transactions on Information Theory* 22.*1* (1976), p. 1-10.

[24] Md Saifur Rahman et Aaron B Wagner, « On the optimality of binning for distributed hypothesis testing », in : *IEEE Transactions on Information Theory* 58.*10* (2012), p. 6282-6303.

[25] Thomas J Richardson et Rüdiger L Urbanke, « The capacity of low-density parity-check codes under message-passing decoding », in : *IEEE Transactions on information theory* 47.*2* (2001), p. 599-618.

[26] Thomas J Richardson, Mohammad Amin Shokrollahi et Rüdiger L Urbanke, « Design of capacity-approaching irregular low-density parity-check codes », in : *IEEE transactions on information theory* 47.*2* (2001), p. 619-637.

[27] David JC MacKay et Radford M Neal, « Near Shannon limit performance of low density parity check codes », in : *Electronics Letters* 32.*18* (1996), p. 1645.

[28] Jeongseok Ha, Jaehong Kim et Steven W McLaughlin, « Rate-compatible puncturing of low-density parity-check codes », in : *IEEE Transactions on Information Theory* 50.*11* (2004), p. 2824-2836.

[29] Dariush Divsalar et al., « Protograph based LDPC codes with minimum distance linearly growing with block size », in : *IEEE Global Telecommunications Conference*, t. 3, IEEE, 2005, 5-pp.

[30] Tom Richardson, Rüdiger Urbanke et al., « Multi-edge type LDPC codes », in : *Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California*, Citeseer, 2002, p. 24-25.

[31] Hironori Uchikawa, « Design of non-precoded protograph-based LDPC codes », in : *IEEE International Symposium on Information Theory*, IEEE, 2014, p. 2779-2783.

[32] Frank R Kschischang, Brendan J Frey et H-A Loeliger, « Factor graphs and the sum-product algorithm », in : *IEEE Transactions on Information Theory* 47.*2* (2001), p. 498-519.

[33] Marc PC Fossorier, Miodrag Mihaljevic et Hideki Imai, « Reduced complexity iterative decoding of low-density parity check codes based on belief propagation », in : *IEEE Transactions on Communications* 47.*5* (1999), p. 673-680.

[34] Jinghu Chen et Marc PC Fossorier, « Density evolution for two improved BP-based decoding algorithms of LDPC codes », in : *IEEE Communications Letters* 6.*5* (2002), p. 208-210.

[35] Jinghu Chen et al., « Reduced-complexity decoding of LDPC codes », in : *IEEE Transactions on Communications* 53.*8* (2005), p. 1288-1299.

[36] Mohamed Yaoumi, « Energy modeling and optimization of protograph-based LDPC codes », thèse de doct., Ecole nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire, 2020.

[37] Christiane L. Kameni Ngassa et al., « Density evolution and functional threshold for the noisy min-sum decoder », in : *CoRR* abs/1405.6594 (2014), arXiv : 1405.6594, URL : http://arxiv.org/abs/1405.6594.

[38] Matteo Gorgoglione, Valentin Savin et David Declercq, « Optimized puncturing distributions for irregular non-binary LDPC codes », in : *International Symposium On Information Theory & Its Applications*, IEEE, 2010, p. 400-405.

[39] Masoud Ardakani et Frank R Kschischang, « A more accurate one-dimensional analysis and design of irregular LDPC codes », in : *IEEE Transactions on Communications* 52.*12* (2004), p. 2106-2114.

[40] Gianluigi Liva et Marco Chiani, « Protograph LDPC codes design based on EXIT analysis », in : *IEEE Global Telecommunications Conference*, IEEE, 2007, p. 3250-3254.

[41] François Leduc-Primeau et Warren J Gross, « Finite-length quasi-synchronous LDPC decoders », in : *International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, IEEE, 2016, p. 325-329.

[42] Raman Yazdani et Masoud Ardakani, « Waterfall performance analysis of finite-length LDPC codes on symmetric channels », in : *IEEE transactions on communications* 57.*11* (2009), p. 3183-3187.

[43] Trevor Hastie et al., *The elements of statistical learning : data mining, inference, and prediction*, t. 2, Springer, 2009.

[44] Jeremy Nadal et al., « Energy optimization of faulty quantized Min-Sum LDPC decoders », in : *International Symposium on Topics in Coding (ISTC)*, IEEE, 2023, p. 1-5.

[45] Rainer Storn et Kenneth Price, « Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces », in : *Journal of global optimization* 11.*4* (1997), p. 341-359.

[46] Xiao-Yu Hu, Evangelos Eleftheriou et Dieter-Michael Arnold, « Regular and irregular progressive edge-growth Tanner graphs », in : *IEEE Transactions on Information Theory* 51.*1* (2005), p. 386-398.

[47] Marc PC Fossorier, « Quasicyclic low-density parity-check codes from circulant permutation matrices », in : *IEEE Transactions on Information Theory* 50.*8* (2004), p. 1788-1793.

[48] Charly Poulliat, Marc Fossorier et David Declercq, « Design of regular (2,dc)-LDPC codes over GF (q) using their binary images », in : *IEEE Transactions on Communications* 56.*10* (2008), p. 1626-1635.

[49] Thomas M Cover, *Elements of information theory*, John Wiley & Sons, 1999.

[50] Yashas Rai, Jesús Gutiérrez et Patrick Le Callet, « A dataset of head and eye movements for 360 degree images », in : *ACM on Multimedia Systems Conference*, 2017, p. 205-210.

[51] Xavier Corbillon et al., « Viewport-adaptive navigable 360-degree video delivery », in : *IEEE International Conference on Communications (ICC)*, IEEE, 2017, p. 1-7.

[52] Michael Gastpar, Pier Luigi Dragotti et Martin Vetterli, « The distributed Karhunen–Loeve transform », in : *IEEE Transactions on Information Theory* 52.*12* (2006), p. 5177-5196.

[53] Francesca Bassi, Michel Kieffer et Claudio Weidmann, « Source coding with intermittent and degraded side information at the decoder », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2008, p. 2941-2944.

[54] Francesca Bassi et al., « Rate-distortion bounds for Wyner–Ziv coding with Gaussian scale mixture correlation noise », in : *IEEE Transactions on Information Theory* 60.*12* (2014), p. 7540-7546.

[55] Elsa Dupraz et al., « Distributed coding of sources with bursty correlation », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2012, p. 2973-2976.

[56] Imre Csiszar, « Linear codes for sources and source networks : Error exponents, universal coding », in : *IEEE Transactions on Information Theory* 28.*4* (1982), p. 585-592.

[57] Rudolf Ahlswede, « Coloring hypergraphs : A new approach to multi-user source coding », in : *Journal of Combinatorics* 4.*1* (1979), p. 76-115.

[58] Rudolf Ahlswede, « Coloring hypergraphs : A new approach to multi-user source coding II », in : *Journal of Combinatorics* 5.*3* (1980), p. 220-268.

[59] H Koga et al., *Information-spectrum methods in information theory*, t. 50, Springer Science & Business Media, 2013.

[60] Aaron D Wyner, « The rate-distortion function for source coding with side information at the decoder : General sources », in : *Information and control* 38.*1* (1978), p. 60-80.

[61] Yasutada Oohama, « Gaussian multiterminal source coding », in : *IEEE Transactions on Information Theory* 43.*6* (1997), p. 1912-1923.

[62] Michael Gastpar, « The Wyner-Ziv problem with multiple sources », in : *IEEE Transactions on Information Theory* 50.*11* (2004), p. 2762-2768.

[63] Elsa Dupraz et al., « Rate-storage regions for extractable source coding with side information », in : *Physical Communication* 37 (2019), p. 100845.

[64] Stark C Draper, « Universal incremental Slepian-Wolf coding », in : *Allerton Conference on Communication, Control and Computing*, Citeseer, 2004, p. 1332-1341.

[65] Roy Timo, Terence Chan et Alex Grant, « Rate distortion with side-information at many decoders », in : *IEEE Transactions on Information Theory* 57.*8* (2011), p. 5240-5257.

[66] Chao Tian et Suhas N Diggavi, « On multistage successive refinement for Wyner–Ziv source coding with degraded side informations », in : *IEEE Transactions on Information Theory* 53.*8* (2007), p. 2946-2960.

[67] Roy Timo, Tobias J Oechtering et Michele Wigger, « Source coding problems with conditionally less noisy side information », in : *IEEE Transactions on Information Theory* 60.*9* (2014), p. 5516-5532.

[68] T.S. Han, *Information-spectrum methods in information theory*, Springer, 2003.

[69] Javier Garcia-Frias, « Compression of correlated binary sources using Turbo codes », in : *IEEE Communications Letters* 5.*10* (2001), p. 417-419.

[70] Anne Aaron et Bernd Girod, « Compression with side information using Turbo codes », in : *Data Compression Conference (DCC)*, IEEE, 2002, p. 252-261.

[71] Angelos D Liveris, Zixiang Xiong et Costas N Georghiades, « Compression of binary sources with side information at the decoder using LDPC codes », in : *IEEE Communications Letters* 6.*10* (2002), p. 440-442.

[72] Elsa Dupraz, Aline Roumy et Michel Kieffer, « Source coding with side information at the decoder and uncertain knowledge of the correlation », in : *IEEE Transactions on Communications* 62.*1* (2013), p. 269-279.

[73] Jun Chen, D-k He et Ashish Jagmohan, « The equivalence between Slepian-Wolf coding and channel coding under density evolution », in : *IEEE Transactions on Communications* 57.*9* (2009), p. 2534-2540.

[74] Elsa Dupraz, Valentin Savin et Michel Kieffer, « Density evolution for the design of non-binary low density parity check codes for Slepian-Wolf coding », in : *IEEE Transactions on Communications* 63.*1* (2014), p. 25-36.

[75] Mohammadreza Yazdani et Amir H Banihashemi, « On construction of rate-compatible low-density parity-check codes », in : *IEEE International Conference on Communications (ICC)*, t. 1, IEEE, 2004, p. 430-434.

[76] David Varodayan, Anne Aaron et Bernd Girod, « Rate-adaptive codes for distributed source coding », in : *Signal Processing* 86.*11* (2006), p. 3123-3130.

[77] Thuy Van Nguyen, Aria Nosratinia et Dariush Divsalar, « The design of rate-compatible protograph LDPC codes », in : *IEEE Transactions on Communications* 60.*10* (2012), p. 2841-2850.

[78] Andrew W Eckford et Wei Yu, « Rateless Slepian-Wolf codes », in : *Asilomar Conference on Signals, Systems and Computers*, 2005, p. 1757-1761.

[79] Chao Yu et Gaurav Sharma, « Improved low-density parity check accumulate (LDPCA) codes », in : *IEEE Transactions on Communications* 61.*9* (2013), p. 3590-3599.

[80] Bane Vasic, « High-rate low-density parity check codes based on anti-Pasch affine geometries », in : *IEEE International Conference on Communications (ICC)*, t. 3, IEEE, 2002, p. 1332-1336.

[81] Feng Cen, « Design of degree distributions for LDPCA codes », in : *IEEE Communications Letters* 13.*7* (2009), p. 525-527.

[82] Fangping Ye et al., « Optimized short-length rate-adaptive LDPC codes for Slepian-Wolf source coding », in : *International Conference on Telecommunications (ICT)*, IEEE, 2018, p. 351-355.

[83] Zeina Mheich et Elsa Dupraz, « Short length non-binary rate-adaptive LDPC codes for Slepian-Wolf source coding », in : *IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2018, p. 1-5.

[84] Fangping Ye et al., « Optimized rate-adaptive protograph-based LDPC codes for source coding with side information », in : *IEEE Transactions on Communications* 67.*6* (2019), p. 3879-3889.

[85] Navid Mahmoudian Bidgoli, Thomas Maugey et Aline Roumy, « Intra-coding of 360-degree images on the sphere », in : *Picture Coding Symposium (PCS)*, IEEE, 2019, p. 1-5.

[86] Catarina Brites, João Ascenso et Fernando Pereira, « Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding », in : *PCS*, Citeseer, 2006.

[87] Velotiaray Toto-Zarasoa, Aline Roumy et Christine Guillemot, « Non-uniform source modeling for distributed video coding », in : *European Signal Processing Conference (EUSIPCO)*, IEEE, 2010, p. 1889-1893.

[88] Francesca Bassi, Michel Kieffer et Claudio Weidmann, « Source coding with intermittent and degraded side information at the decoder », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, p. 2941-2944, doi : 10.1109/ICASSP.2008.4518266.

[89] Navid Mahmoudian Bidgoli, Thomas Maugey et Aline Roumy, « Correlation model selection for interactive video communication », in : *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, p. 2184-2188.

[90] Thomas Maugey et al., « Using an exponential power model for Wyner Ziv video coding », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2010, p. 2338-2341.

[91] Fangping Ye et al., « Bit-plane coding in extractable source coding : optimality, modeling, and application to 360° data », in : *IEEE Communications Letters* 25.*5* (2021), p. 1412-1416.

[92] Adrian Voicila et al., « Low-complexity decoding for non-binary LDPC codes in high order fields », in : *IEEE Transactions on Communications* 58.*5* (2010), p. 1365-1375.

[93] Thomas Maugey et al., « Incremental coding for extractable compression in the context of massive random access », in : *IEEE Transactions on Signal and Information Processing over Networks* 6 (2020), p. 251-260.

[94] Yanwei Liu et al., « RD-optimized interactive streaming of multiview video with multiple encodings », in : *Journal of Visual Communication and Image Representation* 21.*5-6* (2010), p. 523-532.

[95] Stark C Draper et Emin Martinian, « Compound conditional source coding, Slepian-Wolf list decoding, and applications to media coding », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2007, p. 1511-1515.

[96] Mai-Quyen Pham et al., « Optimal reference selection for random access in predictive coding schemes », in : *IEEE Transactions on Communications* 68.*9* (2020), p. 5819-5833.

[97] Christoph Pacher et al., « Information reconciliation for continuous-variable quantum key distribution using non-binary low-density parity-check codes », in : *arXiv preprint arXiv :1602.09140* (2016).

[98] Ronald G Dreslinski et al., « Near-threshold computing : Reclaiming moore's law through energy efficient integrated circuits », in : *Proceedings of the IEEE* 98.*2* (2010), p. 253-266.

[99] Luis Alfonso Lastras-Montaño, Ashish Jagmohan et MM Franceschini, « Algorithms for memories with stuck cells », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2010, p. 968-972.

[100] Christoph Roth et al., « Data mapping for unreliable memories », in : *Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2012, p. 679-685.

[101] John Von Neumann, « Probabilistic logics and the synthesis of reliable organisms from unreliable components », in : *Automata studies* 34 (1956), p. 43-98.

[102] Ester Vicario Bravo, Andrea Bonetti et Andreas Burg, « Data-retention-time characterization of gain-cell eDRAMs across the design and variations space », in : *IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2019, p. 1-5.

[103] Leibin Ni et al., « Distributed in-memory computing on binary RRAM crossbar », in : *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 13.*3* (2017), p. 1-18.

[104] Alexandru Amaricai et al., « Probabilistic gate level fault modeling for near and sub-threshold CMOS circuits », in : *Euromicro Conference on Digital System Design*, IEEE, 2014, p. 473-479.

[105] Gunther Auer et al., « D2. 3 : Energy efficiency analysis of the reference systems, areas of improvements and target breakdown », in : *Earth* 20.*10* (2010).

[106] Avhishek Chatterjee et Lav R Varshney, « Energy-reliability limits in nanoscale circuits », in : *Information Theory and Applications Workshop (ITA)*, IEEE, 2016, p. 1-6.

[107] Yongjune Kim et al., « Generalized water-filling for source-aware energy-efficient SRAMs », in : *IEEE Transactions on Communications* 66.*10* (2018), p. 4826-4841.

[108] K. Ganesan et al., « On the total power capacity of regular-LDPC codes with iterative message-passing decoders », in : *IEEE Journal on Selected Areas in Communications* 34.*2* (2016), p. 375-396, issn : 1558-0008.

[109] Lav R Varshney, « Performance of LDPC codes under faulty iterative decoding », in : *IEEE Transactions on Information Theory* 57.*7* (2011), p. 4427-4444.

[110] SM Sadegh Tabatabaei Yazdi, Hyungmin Cho et Lara Dolecek, « Gallager B decoder on noisy hardware », in : *IEEE Transactions on Communications* 61.*5* (2013), p. 1660-1673.

[111] Chu-Hsiang Huang, Yao Li et Lara Dolecek, « Gallager B LDPC decoder with transient and permanent errors », in : *IEEE Transactions on Communications* 62.*1* (2013), p. 15-28.

[112] Christiane Kameni Ngassa et al., « Density evolution and functional threshold for the noisy min-sum decoder », in : *IEEE Transactions on Communications* 63.*5* (2015), p. 1497-1509.

[113] Alexios Balatsoukas-Stimming et Andreas Burg, « Density evolution for min-sum decoding of LDPC codes under unreliable message storage », in : *IEEE Communications Letters* 18.*5* (2014), p. 849-852.

[114] François Leduc-Primeau, Frank R Kschischang et Warren J Gross, « Modeling and energy optimization of LDPC decoder circuits with timing violations », in : *IEEE Transactions on Communications* 66.*3* (2017), p. 932-946.

[115] David Declercq et al., « Noise-aided gradient descent bit-flipping decoders approaching maximum likelihood decoding », in : *International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, IEEE, 2016, p. 300-304.

[116] Franklin Cochachin et al., « Density evolution thresholds for noise-against-noise min-sum decoders », in : *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2017, p. 1-7.

[117] Christiane L Kameni Ngassa, Valentin Savin et David Declercq, « Unconventional behavior of the noisy min-sum decoder over the binary symmetric channel », in : *Information Theory and Applications Workshop (ITA)*, IEEE, 2014, p. 1-10.

[118] Mohamed Yaoumi et al., « Energy optimization of quantized min-sum decoders for protograph-based LDPC codes », in : *Annals of Telecommunications* 75 (2020), p. 615-621.

[119] Emil Janulewicz et Amir H Banihashemi, « Performance analysis of iterative decoding algorithms with memory over memoryless channels », in : *IEEE Transactions on Communications* 60.*12* (2012), p. 3556-3566.

[120] Elsa Dupraz et Mohamed Yaoumi, « Self-corrected belief-propagation decoder for source coding with unknown source statistics », in : *IEEE Communications Letters* 25.*7* (2021), p. 2133-2137.

[121] Elsa Dupraz et al., « Finite alphabet iterative decoders robust to faulty hardware : Analysis and selection », in : *International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, IEEE, 2014, p. 107-111.

[122] Elsa Dupraz et al., « Analysis and design of finite alphabet iterative decoders robust to faulty hardware », in : *IEEE Transactions on Communications* 63.*8* (2015), p. 2797-2809.

[123] Naveen Verma et Anantha P Chandrakasan, « A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy », in : *IEEE Journal of Solid-State Circuits* 43.*1* (2008), p. 141-149.

[124] Elsa Dupraz et François Leduc-Primeau, « Noisy density evolution with asymmetric deviation models », in : *IEEE Transactions on Communications* 69.*3* (2020), p. 1403-1416.

[125] Elsa Dupraz, David Declercq et Bane Vasic, « Asymptotic error probability of the Gallager B decoder under timing errors », in : *IEEE Communications Letters* 21.*4* (2017), p. 698-701.

[126] Srdan Brkic et al., « On fault tolerance of the Gallager B decoder under data-dependent gate failures », in : *IEEE Communications Letters* 19.*8* (2015), p. 1299-1302.

[127] Eran Sharon, Simon Litsyn et Jacob Goldberger, « Efficient serial message-passing schedules for LDPC decoding », in : *IEEE Transactions on Information Theory* 53.*11* (2007), p. 4076-4091.

[128] Elsa Dupraz, François Leduc-Primeau et François Gagnon, « Low-latency LDPC decoding achieved by code and architecture co-design », in : *IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC)*, IEEE, 2018, p. 1-5.

[129] Christoph Studer et al., « Configurable high-throughput decoder architecture for quasi-cyclic LDPC codes », in : *2008 42nd Asilomar Conference on Signals, Systems and Computers*, IEEE, 2008, p. 1137-1142.

[130] Cédric Marchand, Laura Conde-Canencia et Emmanuel Boutillon, « Architecture and finite precision optimization for layered LDPC decoders », in : *Journal of Signal Processing Systems* 65 (2011), p. 185-197.

[131] Jérémy Nadal et al., « A deeply pipelined, highly parallel and flexible LDPC decoder », in : *2020 18th IEEE International New Circuits and Systems Conference (NEWCAS)*, IEEE, 2020, p. 263-266.

[132] Emmanuel Boutillon et al., « Hardware design and realization for iteratively decodable codes », in : *Channel coding : Theory, algorithms, and applications*, Elsevier, 2014, p. 583-642.

[133] Frank Thomson Leighton, « A graph coloring algorithm for large scheduling problems », in : *Journal of research of the national bureau of standards* 84.*6* (1979), p. 489.

[134] Alfio Di Mauro et al., « Always-on 674μ W@ 4GOP/s error resilient binary neural networks with aggressive SRAM voltage scaling on a 22-nm IoT end-node », in : *IEEE Transactions on Circuits and Systems I : Regular Papers* 67.*11* (2020), p. 3905-3918.

[135] Jérémy Nadal et al., « Towards an Accurate High-Level Energy Model for LDPC Decoders », in : *2021 11th International Symposium on Topics in Coding (ISTC)*, IEEE, 2021, p. 1-5.

[136] Jad Hachem et al., « Coding with encoding uncertainty », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2013, p. 276-280.

[137] Elsa Dupraz et David Declercq, « Evaluation of the robustness of LDPC encoders to hardware noise », in : *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, IEEE, 2015, p. 87-91.

[138] Elsa Dupraz et al., « Practical LDPC encoders robust to hardware errors », in : *IEEE International Conference on Communications (ICC)*, IEEE, 2016, p. 1-6.

[139] Yaoqing Yang, Pulkit Grover et Soummya Kar, « Computing linear transformations with unreliable components », in : *IEEE Transactions on Information Theory* 63.*6* (2017), p. 3729-3756.

[140] Hao Chen, Lav R Varshney et Pramod K Varshney, « Noise-enhanced information systems », in : *Proceedings of the IEEE* 102.*10* (2014), p. 1607-1621.

[141] Michael G Taylor, « Reliable computation in computing systems designed from unreliable components », in : *Bell System Technical Journal* 47.*10* (1968), p. 2339-2366.

[142] Péter Gács et Anna Gál, « Lower bounds for the complexity of reliable Boolean circuits with noisy gates », in : *IEEE Transactions on Information Theory* 40.*2* (1994), p. 579-583.

[143] Nicholas Pippenger, George D Stamoulis et John N Tsitsiklis, « On a lower bound for the redundancy of reliable networks with noisy gates », in : *IEEE Transactions on Information Theory* 37.*3* (1991), p. 639-643.

[144] William Schultz Evans, *Information theory and noisy computation*, t. 94, 57, Citeseer, 1994.

[145] Yaoqing Yang, Pulkit Grover et Soummya Kar, « Fault-tolerant distributed logistic regression using unreliable components », in : *Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2016, p. 940-947.

[146] Michel Verhaegen et Paul Van Dooren, « Numerical aspects of different Kalman filter implementations », in : *IEEE Transactions on Automatic Control* 31.*10* (1986), p. 907-917.

[147] Shuli Sun et al., « Quantized kalman filtering », in : *IEEE International Symposium on Intelligent Control*, IEEE, 2007, p. 7-12.

[148] Di Li et al., « Distributed Kalman filtering with quantized sensing state », in : *IEEE Transactions on Signal Processing* 63.*19* (2015), p. 5180-5193.

[149] Vinay Joshi et al., « Accurate deep neural network inference using computational phase-change memory », in : *Nature Communications* 11.*1* (2020), p. 2473.

[150] Lawrence R Rabiner, « A tutorial on hidden Markov models and selected applications in speech recognition », in : *Proceedings of the IEEE* 77.*2* (1989), p. 257-286.

[151] Erik Ordentlich et Tsachy Weissman, « On the optimality of symbol-by-symbol filtering and denoising », in : *IEEE Transactions on Information Theory* 52.*1* (2005), p. 19-40.

[152] Elsa Dupraz et Lav R Varshney, « Binary recursive estimation on noisy hardware », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2019, p. 877-881.

[153] Maelic Louart et al., « Detection of AIS messages falsifications and spoofing by checking messages compliance with TDMA protocol », in : *Digital Signal Processing* 136 (2023), p. 103983.

[154] Xiaozheng Lai et al., « IoT implementation of Kalman filter to improve accuracy of air quality monitoring and prediction », in : *Applied Sciences* 9.*9* (2019), p. 1831.

[155] Kwangjae Sung et Hwangnam Kim, « Simplified KF-based energy-efficient vehicle positioning for smartphones », in : *Journal of Communications and Networks* 22.*2* (2020), p. 93-107.

[156] Jonathan Kern et al., « Improving the energy-efficiency of a Kalman filter using unreliable memories », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, p. 5345-5349.

[157] Jonathan Kern et al., « Optimizing the energy efficiency of unreliable memories for quantized Kalman filtering », in : *Sensors* 22.*3* (2022), p. 853.

[158] Charles H Roth Jr et Lizy K John, *Digital systems design using VHDL*, Cengage Learning, 2016.

[159] Shubham Jain et al., « Computing in memory with spin-transfer torque magnetic RAM », in : *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.*3* (2017), p. 470-483.

[160] Manuel Le Gallo et Abu Sebastian, « An overview of phase-change memory device physics », in : *Journal of Physics D : Applied Physics* 53.*21* (2020), p. 213002.

[161] Abu Sebastian et al., « Memory devices and applications for in-memory computing », in : *Nature Nanotechnology* 15.*7* (2020), p. 529-544.

[162] Qiuwen Lou et al., « Embedding error correction into crossbars for reliable matrix vector multiplication using emerging devices », in : *ACM/IEEE International Symposium on Low Power Electronics and Design*, 2020, p. 139-144.

[163] Manuel Le Gallo et al., « Precision of bit slicing with in-memory computing based on analog phase-change memory crossbars », in : *Neuromorphic Computing and Engineering* 2.*1* (2022), p. 014009.

[164] Zehui Chen, Clayton Schoeny et Lara Dolecek, « Pilot assisted adaptive thresholding for sneak-path mitigation in resistive memories with failed selection devices », in : *IEEE Transactions on Communications* 68.*1* (2019), p. 66-81.

[165] Elsa Dupraz et Lav R Varshney, « Noisy in-memory recursive computation with memristor crossbars », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2020, p. 804-809.

[166] An Chen, « A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics », in : *IEEE Transactions on Electron Devices* 60.*4* (2013), p. 1318-1326.

[167] Miao Hu et al., « Memristor crossbar based hardware realization of BSB recall function », in : *International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2012, p. 1-7.

[168] Zehui Chen, Clayton Schoeny et Lara Dolecek, « Hamming distance computation in unreliable resistive memory », in : *IEEE Transactions on Communications* 66.*11* (2018), p. 5013-5027.

[169] Elsa Dupraz, Lav R Varshney et François Leduc-Primeau, « Power-efficient deep neural networks with noisy memristor implementation », in : *IEEE Information Theory Workshop (ITW)*, IEEE, 2021, p. 1-5.

[170] Jonathan Kern et al., « MemSE : fast MSE prediction for noisy memristor-based DNN accelerators », in : *IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, IEEE, 2022, p. 62-65.

[171] Natesh S Pillai et Xiao-Li Meng, « An unexpected encounter with Cauchy and Lévy », in : (2016).

[172] Howard Seltman, « Approximations for mean and variance of a ratio », in : *unpublished note* (2012).

[173] Khaled Alhaj Ali et al., « Memristive computational memory using memristor overwrite logic (MOL) », in : *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28.*11* (2020), p. 2370-2382.

[174] Shahar Kvatinsky et al., « Memristor-based material implication (IMPLY) logic : design principles and methodologies », in : *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.*10* (2013), p. 2054-2066.

[175] Nishil Talati et al., « Logic design within memristive memories using memristor-aided logic (MAGIC) », in : *IEEE Transactions on Nanotechnology* 15.*4* (2016), p. 635-650.

[176] Pierre-Emmanuel Gaillardon et al., « The programmable logic-in-memory (PLiM) computer », in : *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Ieee, 2016, p. 427-432.

[177] Dimitrios Stathis, Ioannis Vourkas et Georgios Ch Sirakoulis, « Shortest path computing using memristor-based circuits and cellular automata », in : *International Conference on Cellular Automata*, Springer, 2014, p. 398-407.

[178] Jack Kendall et al., « Training end-to-end analog neural networks with equilibrium propagation », in : *arXiv preprint arXiv :2006.01981* (2020).

[179] Ron M. Roth, « Analog error-correcting codes », in : *IEEE Transactions on Information Theory* 66.*7* (2020), p. 4075-4088.

[180] Deniz Gündüz et al., « Beyond transmitting bits : Context, semantics, and task-oriented communications », in : *IEEE Journal on Selected Areas in Communications* 41.*1* (2022), p. 5-41.

[181] Mostafa El Gamal et Lifeng Lai, « Are Slepian-Wolf rates necessary for distributed parameter estimation ? », in : *Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2015, p. 1249-1255.

[182] Maxim Raginsky, « Learning from compressed observations », in : *IEEE Information Theory Workshop*, IEEE, 2007, p. 420-425.

[183] Philippe Mary Jiahui Wei Elsa Dupraz, « Asymptotic and non-asymptotic rate-loss bounds for linear regression with side information », in : *European Signal Processing Conference (EUSIPCO)*, 2023, p. 1-5.

[184] Naftali Tishby et Noga Zaslavsky, « Deep learning and the information bottleneck principle », in : *IEEE Information Theory Workshop (ITW)*, IEEE, 2015, p. 1-5.

[185] Sreejith Sreekumar et Deniz Gündüz, « Distributed hypothesis testing over discrete memoryless channels », in : *IEEE Transactions on Information Theory* 66.*4* (2019), p. 2044-2066.

[186] Sadaf Salehkalaibar, Michèle Wigger et Ligong Wang, « Hypothesis testing over the two-hop relay network », in : *IEEE Transactions on Information Theory* 65.*7* (2019), p. 4411-4433.

[187] Gil Katz et al., « On the necessity of binning for the distributed hypothesis testing problem », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2015, p. 2797-2801.

[188] Gil Katz, Pablo Piantanida et Mérouane Debbah, « Distributed binary detection with lossy data compression », in : *IEEE Transactions on Information Theory* 63.*8* (2017), p. 5207-5227.

[189] Ertem Tuncel et Deniz Gündüz, « Identification and lossy reconstruction in noisy databases », in : *IEEE Transactions on Information Theory* 60.*2* (2013), p. 822-831.

[190] Mark A Davenport et al., « Signal processing with compressive measurements », in : *IEEE Journal of Selected topics in Signal Processing* 4.*2* (2010), p. 445-460.

[191] Dominique Pastor et Francois-Xavier Socheleau, « Random distortion testing with linear measurements », in : *Signal Processing* 145 (2018), p. 116-126.

[192] Elsa Dupraz, Dominique Pastor et François-Xavier Socheleau, « A statistical signal processing approach to clustering over compressed data », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, p. 2791-2795, doi : 10.1109/ICASSP.2018.8462355.

[193] Farhad Pourkamali-Anaraki et Stephen Becker, « Preconditioned data sparsification for big data with applications to PCA and K-means », in : *IEEE Transactions on Information Theory* 63.*5* (2017), p. 2954-2974.

[194] David L Donoho, « Compressed sensing », in : *IEEE Transactions on Information Theory* 52.*4* (2006), p. 1289-1306.

[195] Elsa Dupraz, « K-means algorithm over compressed binary data », in : *Data compression conference (DCC)*, 2018, doi : 10.1109/DCC.2018.00060.

[196] Max Ehrlich et Larry S Davis, « Deep residual learning in the JPEG transform domain », in : *IEEE International Conference on Computer Vision*, 2019, p. 3484-3493.

[197] Rémi Piau, Thomas Maugey et Aline Roumy, « Learning on entropy coded images with cnn », in : *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, p. 1-5.

[198] Alireza Tasdighi et Elsa Dupraz, « An End-to-End Scheme for Learning Over Compressed Data Transmitted Through a Noisy Channel », in : *IEEE Access* 11 (2023), p. 8254-8267.

[199] Mikolaj Jankowski, Deniz Gündüz et Krystian Mikolajczyk, « Wireless image retrieval at the edge », in : *IEEE Journal on Selected Areas in Communications* 39.*1* (2020), p. 89-100.

[200] Yashas Malur Saidutta, Afshin Abdi et Faramarz Fekri, « Analog joint source-channel coding for distributed functional compression using deep neural networks », in : *IEEE International Symposium on Information Theory (ISIT)*, 2021, p. 2429-2434.

[201] Chia-Han Lee et al., « Deep learning-constructed joint transmission-recognition for Internet of Things », in : *IEEE Access* 7 (2019), p. 76547-76561.

[202] Kart-Leong Lim, Xudong Jiang et Chenyu Yi, « Deep clustering with variational autoencoder », in : *IEEE Signal Processing Letters* 27 (2020), p. 231-235.

[203] Victoria Kostina et Sergio Verdú, « Lossy joint source-channel coding in the finite blocklength regime », in : *IEEE Transactions on Information Theory* 59.*5* (2013), p. 2545-2575.

[204] Reinhard Heckel, Gediminas Mikutis et Robert N. Grass, « A characterization of the DNA data storage channel », en, in : *Scientific Reports* 9.*1* (2019), p. 9663, issn : 2045-2322, (visité le 08/10/2020).

[205] Dominique Lavenier, « DNA storage : Synthesis and sequencing semiconductor technologies », in : *IEDM 2022 - 68th Annual IEEE International Electron Devices Meeting*, San Francisco, United States : IEEE, déc. 2022, p. 1-4, url : `https://hal.science/hal-03902786`.

[206] Cheng Kai Lim et al., « Novel modalities in DNA data storage », in : *Trends in Biotechnology* 39.*10* (2021), p. 990-1003.

[207] Yunhao Wang et al., « Nanopore sequencing technology, bioinformatics and applications », in : *Nature Biotechnology* 39.*11* (2021), p. 1348-1365.

[208] Belaid Hamoum et Elsa Dupraz, « Channel model and decoder with memory for DNA data storage with nanopore sequencing », in : *IEEE Access* (2023).

[209] VI Levenshtein, « Asymptotically optimum binary code with correction for losses of one or two adjacent bits », in : *Problemy Kibernetiki* 19 (1967), p. 293-298.

[210] Feng Wang, Dario Fertonani et Tolga M Duman, « Symbol-level synchronization and LDPC code design for insertion/deletion channels », in : *IEEE Transactions on Communications* 59.*5* (2011), p. 1287-1297.

[211] A.S.J. Helberg et H.C. Ferreira, « On multiple insertion/deletion correcting codes », in : *IEEE Transactions on Information Theory* 48.*1* (2002), p. 305-308, doi : `10.1109/18.971760`.

[212] Ryo Shibata, Gou Hosoya et Hiroyuki Yashima, « Design of irregular LDPC codes without markers for insertion/deletion channels », in : *IEEE Global Communications Conference*, IEEE, 2019, p. 1-6.

[213] Andreas Lenz et al., « Concatenated codes for recovery from multiple reads of DNA sequences », in : *IEEE Information Theory Workshop (ITW)*, 2021, p. 1-5.

[214] Dominique Lavenier, « Constrained consensus sequence algorithm for DNA archiving », in : *CoRR* abs/2105.04993 (2021), arXiv : 2105.04993, URL : https://arxiv.org/abs/2105.04993.

[215] Belaid Hamoum et al., « Channel model with memory for DNA data storage with nanopore sequencing », in : *International Symposium on Topics in Coding (ISTC)*, IEEE, 2021, p. 1-5.

[216] Yu Li et al., « DeepSimulator1. 5 : a more powerful, quicker and lighter simulator for Nanopore sequencing », in : *Bioinformatics* 36.*8* (2020), p. 2578-2580.

[217] Yukiteru Ono, Kiyoshi Asai et Michiaki Hamada, « PBSIM2 : a simulator for long-read sequencers with a novel generative model of quality scores », in : *Bioinformatics* 37.*5* (2021), p. 589-595.

[218] Ryan R Wick, « Badread : simulation of error-prone long reads », in : *Journal of Open Source Software* 4.*36* (2019), p. 1316.

[219] Belaıd Hamoum, Aref Ezzeddine et Elsa Dupraz, « Synchronization algorithms from high-rate LDPC codes for DNA data storage », in : *International Conference on Digital Signal Processing (DSP)*, IEEE, 2023, p. 1-5.

[220] Reinhard Heckel et al., « Fundamental limits of DNA storage systems », in : *IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2017, p. 3130-3134.

[221] Andreas Lenz et al., « An upper bound on the capacity of the DNA storage channel », in : *IEEE Information Theory Workshop (ITW)*, IEEE, 2019, p. 1-5.

[222] Nir Weinberger et Neri Merhav, « The DNA storage channel : Capacity and error probability bounds », in : *IEEE Transactions on Information Theory* 68.*9* (2022), p. 5657-5700.

[223] Yixin Wang et al., « Construction of bio-constrained code for DNA data storage », in : *IEEE Communications Letters* 23.*6* (2019), p. 963-966.

[224] Xavier Pic et al., « Rotating labeling of entropy coders for synthetic DNA data storage », in : *International Conference on Digital Signal Processing (DSP)*, IEEE, 2023, p. 1-5.

[225] Chloé Berton, Gouenou Coatrieux et Dominique Lavenier, « A first proposal for secure data storage into DNA molecules compliant with biological constraints », in : *DSMM 2022-1st International Conference on Data Storage in Molecular Media*, 2022.

[226] Benjamin T James, Brian B Luczak et Hani Z Girgis, « MeShClust : an intelligent tool for clustering DNA sequences », in : *Nucleic acids research* 46.*14* (2018), e83-e83.

[227] Congzhou Chen et al., « DNA logic circuits based on accurate step function gate », in : *IEEE Access* 8 (2020), p. 125513-125520.

[228] S Okumura et al., « Nonlinear decision-making with enzymatic neural networks », in : *Nature* 610.*7932* (2022), p. 496-501.