Study of non-binary turbo codes for future communication and broadcasting systems

Rami Klaimi

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Télécommunications*

Par
Rami KLAIMI

Etude de turbocodes non binaires pour les futurs systèmes de communication et de diffusion

Study of non-binary turbo codes for future communication and broadcasting systems (in English)

Thèse présentée et soutenue à Brest, le 3 juillet 2019
Unité de recherche : Lab-STICC
Thèse N° : 2019IMTA0141

Rapporteurs avant soutenance :

Gianluigi Liva	Chercheur au DLR Wessling
Christophe Jégo	Professeur, INP/Enseirb-Matmeca Bordeaux

Composition du Jury :

Président :	Charly Poulliat	Professeur, INP/Enseeiht Toulouse
Rapporteurs :	Gianluigi Liva	Chercheur au DLR Wessling
	Christophe Jégo	Professeur, INP/Enseirb-Matmeca Bordeaux
Examinateur :	Charly Poulliat	Professeur, INP/Enseeiht Toulouse
Directrices de thèse :	Catherine Douillard	Professeure, IMT Atlantique Brest
	Joumana Farah	Professeure, Université Libanaise Roumieh
Encadrant de thèse :	Charbel Abdel Nour	Maître de conférences, IMT Atlantique Brest
Invité :	Emmanuel Boutillon	Professeur, Université Bretagne Sud Lorient

Sous le sceau de l'Université Bretagne Loire

IMT Atlantique

École Doctorale MATHSTIC

Study of non-binary turbo codes for future communication and broadcasting systems

---

Thèse de Doctorat

Spécialité : Télécommunications

Présentée par **Rami Klaimi**

Département : Électronique
Laboratoire : Lab-STICC

Directeurs de thèse : **Catherine Douillard, Joumana Farah**

Encadrant : **Charbel Abdel Nour**

Abstract

Study of non-binary turbo codes for future communication and broadcasting systems

Rami Klaimi

Electronics Department, IMT Atlantique

In July 2018, the International Telecommunication Union launched a new focus group, FG NET-2030, to study the capabilities of networks for the year 2030 and beyond, which are expected to support novel forward-looking scenarios, such as holographic communications, autonomous vehicles, massive machine-type communications, tactile Internet, ... In the study of network 2030 architecture, new communication requirements are expected to emerge that are more sensitive to resource demands, specifically very-high throughput to support explosive bandwidth-intensive future applications beyond the limitation of near future 5G networks. In this context, new error correcting codes targeting very high throughput (Tbps) communications should be investigated. To this end, this thesis investigates a new family of error correcting codes based on non-binary turbo codes defined over Galois fields GF(q), with $q = 2^m$ and $m > 1$. The thesis details the different constituent blocks of the turbo code structure which have been studied towards the definition of promising non-binary turbo codes.

After a short review of the state of the art on the existing channel coding techniques based on non-binary codes, the main contribution of Chapter 2 is dedicated to the proposition of a new structure of non-binary convolutional codes, based on a memory-1 recursive systematic convolutional code defined over GF(q), well suited for the design of turbo codes. The proposed structure has the lowest encoding and decoding complexity in its corresponding field GF(q) and shows promising error correction capabilities. Then, we derive a low-complexity method to estimate the cumulated Euclidean distance spectrum of this family of codes and we provide the results of the search for instances of the best and worst codes according to the distance spectrum criterion. In addition, we introduce a downsizing technique based on the merging process of constellation symbols, targeting the enhancement of the error correction capabilities of convolutional codes, while lowering the demapping complexity. Finally, we propose a method to optimize the non-binary coefficients in order to improve the performance of the proposed codes when mapped to binary constellations.

In Chapter 3, using the non-binary convolutional codes proposed in Chapter 2, we study the complete structure of non-binary turbo codes. We start by analyzing the effect of different interleaving techniques on the performance of these codes, and we propose new constraints to be applied on the well-known Almost Regular Permutation (ARP) interleavers in order to enhance the performance of non-binary turbo codes. Afterwards, we show how different puncturing techniques can be adapted to the non-binary domain. Finally, we propose a method to estimate the cumulated Euclidean distance spectrum of non-binary turbo codes, in order to compute the truncated union bounds expressions in the non-binary domain.

A new structure of non-binary turbo codes is proposed in Chapter 4, where we study the addition of a new block to the conventional structure of turbo codes. This block defines a bijective symbol transformation designed to supplement the interleaver in order

to increase the minimum Euclidean distance of the turbo code. We show that the proposal significantly increases the coding gains of non-binary turbo codes, and very low error floors can be achieved. Performance comparisons of the proposed codes with the best known error correction codes from the literature are conducted. Non-binary turbo codes are shown to outperform the existing codes under different simulation conditions.

Chapter 5 is devoted to the introduction of a low-complexity decoder for non-binary turbo codes. First, we present a literature review about the low-complexity decoding algorithms defined in the context of non-binary low-density parity-check codes. Then, we develop a new simplified decoding algorithm, based on scaled Min-Log-MAP decoding, for the proposed structure of non-binary turbo codes. The proposal is shown to limit the storage requirements and the computational complexity in comparison with the basic decoding algorithm. Simulation results confirm that only low performance losses are introduced, while high complexity reductions are achieved.

Keywords: Convolutional codes, turbo codes, Galois fields, non-binary codes, Euclidean distance spectrum, coded modulation, ARP interleavers, correlation in decoding, bijective transformation, decoding algorithm, complexity reduction.

Contents

List of Figures

List of Tables

Abbreviations

ACS	Add Compare Select operation.
ARP	Almost Regular Permutation.
AWGN	Additive White Gaussian Noise.
BC	Bubble Check.
BCH	Bose-Chaudhuri-Hocquenghem.
BER	Bit Error Rate.
BICM	Bit-interleaved Coded Modulation.
BiP	Bit Puncturing.
BLER	BLock Error Rate.
BP	Belief Propagation.
BPSK	Binary Phase-Shift Keying.
CBRM	Circular Buffer Rate Matching.
CC	Convolutional code.
CM	Coded Modulation.
CN	Check Node.
CSED	Cumulated squared Euclidean distance.
DC	Diverging Converging sequences.
DRP	Dithered Relative Prime.
ECN	Elementary Check Node.
ED	Euclidean distance.
EMS	Extended Min-Sum.
FEC	Forward Error Correction.
FER	Frame Error Rate.
GF	Galois field.
IoT	Internet of Things.
LDPC	Low-Density Parity-Check.
LLR	Log-Likelihood Ratio.
LP	Logarithm of Probability.
LSB	Least Significant Bit.

MAP	Maximum A Posteriori.
ML	Maximum Likelihood.
MLM	Min-Log-MAP.
MSB	Most Significant Bit.
NB	Non-Binary.
NB-CC	Non-Binary Convolutional code.
NB-LDPC	Non-Binary Low-Density Parity-Check.
NB-TC	Non-Binary Turbo code.
PEP	Pairwise Error Probability.
PNC	Physical-Layer Network Coding.
PSK	Phase-Shift Keying.
QAM	Quadrature Amplitude Modulation.
QPP	Quadratic Permutation Polynomial.
QPSK	Quadrature Phase-Shift Keying.
RI	Regular Interleaver.
RSC	Recursive Systematic Convolutional.
RTZ	Return To Zero.
SED	Squared Euclidean distance.
SER	Symbol Error Rate.
SNR	Signal-to-Noise Ratio.
SP	Symbol Puncturing.
TC	Turbo Code.
TUB	Truncated Union Bound.
TWRC	Two-Way Relay Channel.
URLLC	Ultra Reliable Low Latency.
VN	Variable Node.

Notations

α	Forward metric.
β	Backward metric.
γ	Transition metric.
c	Transmitted codeword.
C	Channel capacity.
d	Data sequence.
d'	Interleaved data sequence.
D_{Euc}	Cumulated Euclidean distance between two sequences.
$d_{\min}$	Minimum distance of the code.
E_b	Transmitted energy per information bit.
E_s	Transmitted energy per symbol.
g	Correlation girth.
G	Generator matrix.
H	Parity-Check matrix.
I_x	In-phase component of signal x.
$I(x,r)$	Mutual information.
K	Data sequence length.
$L_i^e(a)$	Extrinsic LLR.
M	Puncturing period.
N	Codeword length.
N_0	Noise power spectral density.
$n(d_i)$	Multiplicity of distance d_i.
n_{it}	Number of performed decoding iterations.
np_p	Number of non-punctured parity symbols in a correlation cycle.
np_s	Number of non-punctured systematic symbols in a correlation cycle.
Π	Interleaving function.

P	Regular interleaver period.
q	Order of the Galois field.
Q	Disorder degree of the ARP interleaver.
Q_x	Quadrature component of signal x.
R	Code rate.
$r(d_i)$	Normalized multiplicity of distance d_i
S	Vector of shifts of the ARP interleaver.
$S_{\min}$	Minimum spread value associated to an interleaver.
T	Bijective transformation connections.
$T_g(\mathbf{d}, \mathbf{d}')$	Number of transitions between $\mathbf{d}$ and $\mathbf{d}'$ in g.

Etude de turbocodes non binaires pour les futurs systèmes de communication et de diffusion

Les systèmes de communication sont devenus un élément clé pour le développement de la société actuelle. Les activités humaines sont de plus en plus dépendantes des évolutions technologiques. Les nouveaux services de communication concernent notamment les services multimédias, les interactions homme-machine et les communications de machine à machine, avec l'apparition de nouveaux scénarios tels que les communications holographiques, la conduite autonome de véhicules ou l'internet tactile. L'expansion de ces nouveaux services va se traduire par une augmentation significative du nombre d'appareils connectés. Ainsi, certaines études telles que [1] ont montré que le nombre total d'appareils connectés dépassera la population mondiale d'ici 2022, pour atteindre environ 12,3 milliards d'appareils. Il est par conséquent nécessaire que les réseaux de télécommunication prennent en charge ces nouveaux scénarios, en permettant des communications fiables et spectralement efficaces afin de pouvoir prendre en charge les applications futures à bande passante élevée, au-delà des limites des réseaux de $5^{\text{ème}}$ génération (5G) à venir. Les standards de communication existants ne sont pas encore adaptés à cette croissance. En particulier, les schémas de modulations codées adoptés dans les normes de communication ne permettent toujours pas d'approcher les limites théoriques de transmission établies par Claude Shannon [2] pour les efficacités spectrales élevées. Pour ce faire, les codes non binaires (NB) ont été identifiés comme une solution possible permettant d'améliorer les performances de correction d'erreurs par rapport aux codes binaires classiques lorsque des modulations d'ordres élevés sont considérées. A cette fin, les travaux de cette thèse portent sur la définition d'une nouvelle famille de codes correcteurs d'erreurs, basée sur les turbocodes non binaires (NB-TC) et répondant mieux aux nouvelles exigences des systèmes de communications. Dans un premier temps, nous étudions une nouvelle structure des codes convolutifs non binaires offrant de meilleures performances que les structures existantes. Ensuite, des méthodes d'entrelacement et de poinçonnage sont proposées pour la construction de NB-TCs. Puis, la structure des turbocodes est modifiée en ajoutant une fonction de transformation permettant l'amélioration des gains de codage de ces codes. Enfin, nous nous intéressons à l'élaboration d'un nouvel algorithme de décodage simplifié pour obtenir des NB-TCs de complexité acceptable pour une mise en œuvre pratique.

Chapitre 1. Techniques de codage canal et codes non binaires

Le premier chapitre présente une synthèse des concepts de codage de canal ainsi que nos motivations pour étudier les codes non binaires.

Techniques de codage canal

Le codage de canal est défini comme l'outil permettant la correction de messages transmis par l'intermédiaire d'un canal de communication sujet aux erreurs. De nombreux codes correcteurs d'erreurs ont été introduits dans la littérature depuis les travaux de Shannon à la fin des années 1940. Une avancée capitale dans le domaine a eu lieu en 1993 avec l'introduction des turbocodes (TC) par Claude Berrou *et al.* [3], définis par la concaténation parallèle de deux ou plusieurs encodeurs élémentaires. Les TCs ont été adoptés pour les standards de 3$^{\text{ème}}$ et 4$^{\text{ème}}$ générations de communications mobiles, notamment le standard actuel *long term evolution* (LTE), tandis que les codes concurrents *low-density parity-check* (LDPC) [4] et les codes polaires [5] ont été adoptés pour la 5G.

Cependant, avec l'avènement de la 5G, de nouvelles exigences sont apparues. Pour les applications nécessitant une faible latence et un débit élevé, par exemple pour des interventions chirurgicales à distance, des codes à très faibles taux d'erreurs sont d'une importance capitale pour éviter la retransmission de paquets. En outre, des codes très robustes sont requis pour la transmission de très courtes trames dans le contexte de scénarios de communications très fiables latence *ultra reliable low latency communications* (URLLC) et de l'internet des objets (IoT). De plus, pour ces dernières applications, une plus longue durée de vie de la batterie de l'appareil doit également être assurée. Ces multiples contraintes font appel à des caractéristiques contradictoires en ce sens que l'amélioration des performances nécessite en général une augmentation de la complexité (et donc de la consommation) et/ou de la latence. Par conséquent, aucun des codes définis jusqu'à présent ne peut satisfaire simultanément l'ensemble de ces nouvelles exigences, et l'introduction d'une nouvelle famille de codes correcteurs d'erreurs, permettant de faire un pas de plus vers la satisfaction simultanée de ces contraintes est d'une importance capitale.

Motivation pour étudier les codes non binaires

Plusieurs motivations nous ont incités à étudier des codes correcteurs d'erreurs dans le domaine non binaire (NB) :

1. Meilleure performance de correction que les codes binaires pour la transmission de blocs courts [6],

2. Du point de vue théorique, les codes NB peuvent offrir de meilleures performances que les codes binaires lorsqu'ils sont associés à des modulations d'ordre élevé comme le montre la Fig. 1,

3. La conception des codes NB offre plus de degrés de liberté que pour les codes binaires car des corps d'ordres élevés sont exploités.

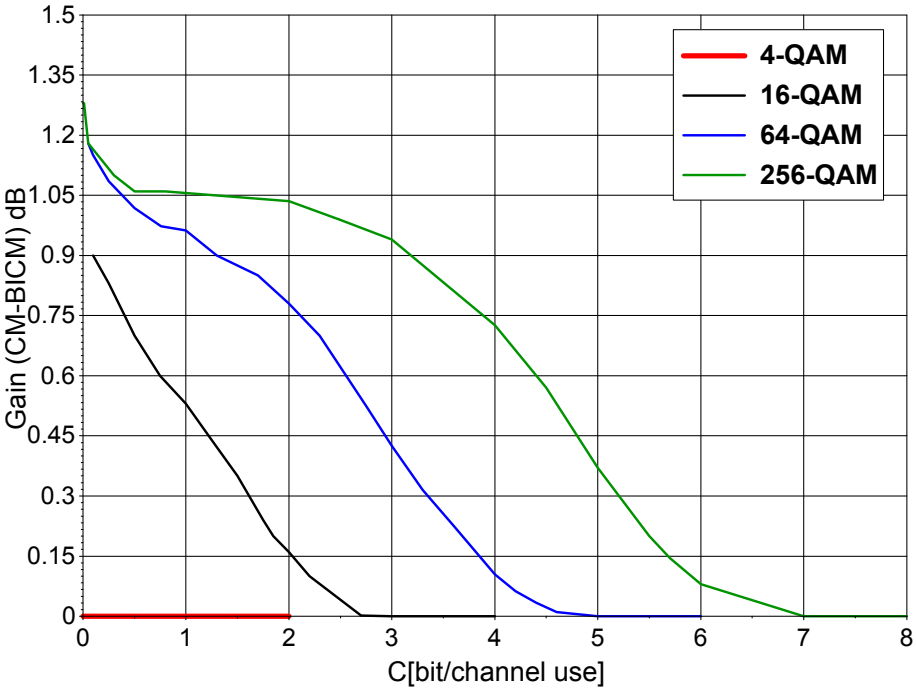

Figure 1: Gain de capacité offert par les codes NB par rapport aux codes binaires en fonction de l'efficacité spectrale, pour différents types de constellations.

Turbocodes

Cette thèse s'intéresse spécifiquement à la famille des NB-TCs. Les TCs, tels qu'introduits dans [3], sont constitués par la concaténation parallèle de deux ou plusieurs codes convolutifs systématiques et récursifs (RSC) séparés par une fonction d'entrelacement ou permutation.

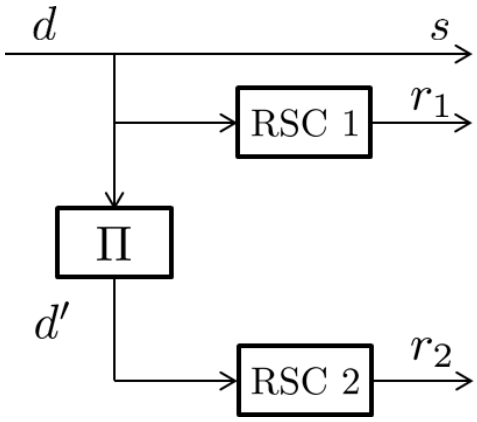

Figure 2: Structure de turbocodage : concaténation en parallèle de deux codes RSC, séparés par un entrelaceur Π.

Comme le montre la Fig. 2, la trame d'information d est encodée par le premier codeur RSC, avant d'être entrelacée par la fonction de permutation Π pour former la version d', encodée par le second codeur RSC.

Codes convolutifs récursifs et systématiques

Un code convolutif peut être vu comme une machine d'états finis composée d'un registre à décalage et des fonctions algébriques linéaires. Il est caractérisé par le nombre d'éléments mémoire ν, qui impose le nombre d'états possibles de l'encodeur, 2^{ν} dans le cas d'un code binaire. Le message de sortie du code convolutif est formé à partir des informations redondantes générées par la structure du code. Un code convolutif est systématique lorsque son message d'entrée est transmis parmi les symboles redondants. Quand la structure du registre à décalage est rétroactive, le code convolutif devient récursif. L'utilisation des codes convolutifs sous une forme récursive s'avère essentielle pour la construction des turbocodes [7], car les codes non récursifs peuvent avoir des motifs d'erreur de poids 1, et dans ce cas la performance des turbocodes ne dépend pas de la taille de trame ni de l'entrelacement. Au contraire, dans le cas des codes récursifs, la performance de correction s'améliore avec l'augmentation de la taille de trame grâce au gain d'entrelacement. Un exemple de code RSC est donné en Fig. 3 : il s'agit du code utilisé dans le standard LTE.

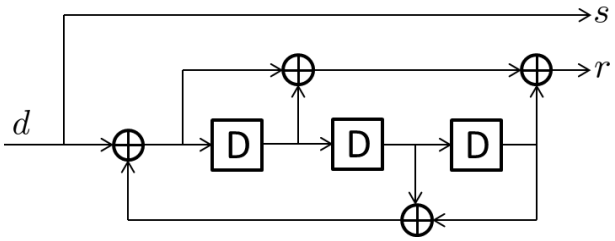

Figure 3: Exemple d'une structure RSC. Les polynômes générateurs sont $(13,15)_8$ utilisés dans la norme LTE.

La fonction d'entrelacement

L'entrelaceur du TC est une fonction de permutation qui réarrange la trame d'information d à l'entrée du codeur élémentaire "RSC 2". Le rôle premier de cet entrelaceur est de lutter efficacement contre l'apparition d'erreurs à l'entrée d'au moins un des décodeurs élémentaires, côté récepteur. Mais la conception de l'entrelaceur a également un impact direct sur la distance de Hamming minimale du TC et donc de sa performance à faible taux d'erreur [8]. Afin d'obtenir une distance minimale élevée, plusieurs contraintes doivent être prises en compte dans la conception de l'entrelaceur [9, 10].

Poinçonnage des turbocodes

La structure du TC de la Fig. 2 produit des mots de code avec un rendement de codage $R = \frac{K}{N} = 1/3$. Des rendements de codage plus élevés mèneront à des débits plus élevés, tandis que des rendements de codage plus faibles permettront d'obtenir de meilleures performances de correction. Selon le type d'application, un rendement de codage particulier est ciblé. Pour obtenir des rendements élevés, la technique la plus couramment utilisée consiste à poinçonner le code [11], c'est-à-dire à ne pas transmettre certains bits codés, systématiques ou de parité.

Chapitre 2. Conception de codes convolutifs sur GF(q)

La première étape vers la construction des NB-TCs est la définition de codes convolutifs NB (NB-CC). Dans ce chapitre, nous proposons une nouvelle structure générale pour ces codes ainsi qu'une méthode permettant d'identifier les meilleurs NB-CCs sur chaque corps de Galois GF(q).

Structures de NB-CCs étudiées

Deux structures de codes convolutifs à un seul élément mémoire sont proposées dans ce travail, comme le montre la Fig. 4. Les deux structures diffèrent de par le calcul du symbole de parité généré. Dans la structure S_1, le symbole de parité est calculé directement à partir de l'état de l'accumulateur : il est donc égal pour toutes les transitions du treillis qui convergent vers un état donné. Au contraire, dans la structure S_2, deux états successifs de l'accumulateur contribuent au calcul du symbole de parité, dont la valeur diffère pour les différentes transitions du treillis convergeant vers un même état. Grâce à l'utilisation d'une structure à un seul élément mémoire, les NB-CCs considérés ont la particularité d'avoir des treillis totalement connectés. Les structures proposées dépendent de deux (structure S_1) ou trois (structure S_2) coefficients dans le corps de Galois correspondant. Plus de coefficients peuvent être ajoutés à ces structures mais nous avons prouvé que l'utilisation de deux coefficients pour S_1 et trois coefficients pour S_2 suffit à couvrir tous les NB-CCs possibles sur GF(q) lorsqu'ils sont associés à une constellation q-QAM.

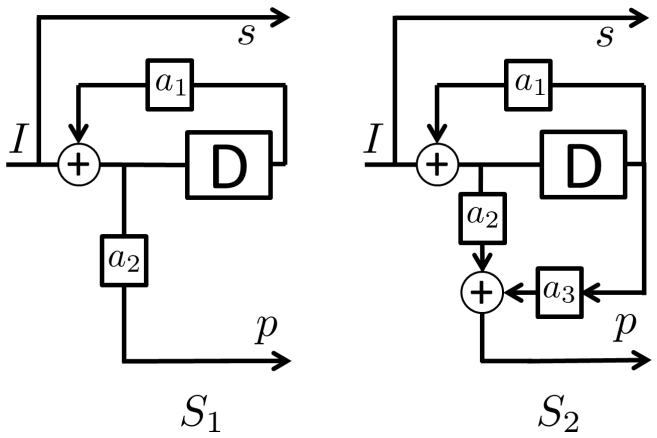

Figure 4: Structures de code convolutifs non binaires avec un élément mémoire.

Sélection des NB-CCs sur chaque corps de Galois

Afin de trouver les meilleurs NB-CC sur chaque corps de Galois, il faut définir le critère principal qui aide à identifier les polynômes générateurs correspondants. Lorsqu'un NB-CC sur GF(q) est associé à une constellation q-QAM, ses performances asymptotiques de correction, c.-à-d. à fort rapport signal à bruit (SNR), dépendent de sa distance euclidienne cumulée (CED) minimale : plus elle est élevée, meilleure est la correction. Ainsi, on cherche à identfier les NB-CCs avec des CED minimales élevées pour chaque corps de Galois. Le treillis des NB-CCs considérés étant totalement connecté, les séquences dans le

treillis à faible distance entre elles (et donc sujettes aux erreurs) ont été identifiées comme les couples de séquences courtes qui divergent d'un état puis convergent à nouveau vers un état. Nous appelons ces paires de séquences des séquences ou chemins DC (divergents et convergents). La CED correspondante est calculée comme la somme des distances euclidiennes entre les symboles transmis le long des deux chemins DC.

La recherche de bons NB-CCs consiste à rechercher l'ensemble des coefficients a_1, a_2 et a_3 (cf. Fig. 4) qui maximisent les valeurs de CED les plus faibles tout en minimisant leurs multiplicités (c.-à-d. le nombre de paires de séquences avec une distance donnée). Nous avons démontré que les NB-CCs ainsi obtenus surpassent les codes convolutifs binaires et les NB-CCs existants dans la littérature en termes de performance de correction d'erreurs.

Optimisation des NB-CCs pour les constellations d'ordre inférieur

Afin d'utiliser des constellations d'ordre inférieur avec les NB-CCs ainsi définis, nous avons proposé de modifier légèrement la structure des codes en ajoutant un nouveau coefficient a_s comme indiqué sur la Fig. 5, afin de maximiser la CED minimale. En ajoutant ce coefficient, on obtient des NB-CCs qui présentent des performances de correction surpassant les codes convolutifs binaires de même complexité, lorsqu'ils sont associés à des constellations binaires.

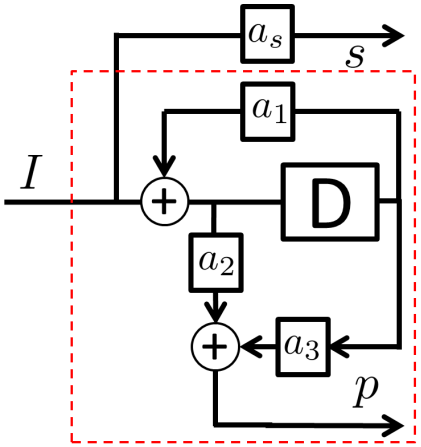

Figure 5: Structure du NB-CC introduisant le nouveau coefficient a_s.

Chapitre 3. Turbocodes non binaires: entrelacement, poinçonnage, borne de l'union

Après avoir construit des NB-CCs avec de bonnes propriétés de distance, l'étape suivante vers la conception de bons NB-TCs définis sur GF(q) consiste à étudier des entrelaceurs et des motifs de poinçonnage qui fournissent des gains de codage élevés dans toutes les régions de SNR. Ce chapitre étudie la conception d'entrelaceurs et des motifs de poinçonnage efficaces pour les NB-TC. Il traite également le calcul des bornes de l'union pour les NB-TCs afin de prédire leur comportement asymptotique.

Influence des contraintes d'entrelacement sur les NB-TCs

Effet d'une faible valeur de *spread*

Le *spread* est défini comme la valeur minimale de la somme des distances spatiales entre deux symboles d'information dans l'ordre naturel (c.-à-d. avant entrelacement) et dans l'ordre entrelacé. En appliquant un schéma d'entrelacement avec des valeurs de *spread* trop faibles, il arrive souvent que deux symboles voisins dans la trame non entrelacée soient toujours proches dans la trame entrelacée. En raison de la structure de CC adoptée, cela induit des séquences DC de longueur 2 avec la même faible valeur de CED pour les deux codes constitutifs. Dans de telles conditions, il est fort probable que des séquences erronées dans un décodeur ne puissent pas être corrigées par l'autre décodeur, ce qui entraîne un taux d'erreur élevé. Il faut par conséquent veiller à ce que l'entrelaceur soit construit de telle sorte à garantir des valeurs de *spread* suffisamment grandes.

Effet des cycles de corrélation courts

Nous avons observé que la présence de cycles de corrélation courts dans le schéma d'entrelacement crée des séquences DC composites (c.-à-d. constituées de plusieurs séquences DC élémentaires) à faibles CEDs. Pour limiter la corrélation dans le décodeur et obtenir des valeurs minimales de CED élevées, il est préférable de mettre en œuvre des entrelaceurs structurés, tels que les entrelaceurs *almost regular permutation* (ARP) qui sont déjà largement utilisés pour les TC binaires. L'entrelaceur ARP adopté dans la thèse suit le modèle donné dans [12]:

$$\Pi(i) = (P.i + S(i \bmod Q)) \bmod K \tag{1}$$

où $i = 0 \ldots K-1$ représente l'adresse entrelacée du symbole de données et $\Pi(i)$ l'adresse correspondante dans l'ordre naturel. P représente la période de l'entrelaceur régulier correspondant, Q est le degré de désordre et S définit le vecteur des Q valeurs de décalage.

Réduire l'effet de corrélation pour les NB-TCs utilisant l'entrelacement ARP

Dans le cas des TCs binaires, il a été montré que l'introduction du modèle d'entrelaceur ARP permet à la fois d'obtenir des valeurs de *spread* élevées, tout en augmentant la longueur minimale des cycles de corrélation (grandeur encore appelée *girth*) [13]. Toutefois, lorsque les règles de construction des entrelaceurs ARP binaires sont appliquées, les cycles de corrélation les plus courts restent problématiques dans le processus de décodage des NB-TCs. Nous avons par conséquent proposé plusieurs nouvelles contraintes à appliquer dans le cas NB pour cibler les paramètres d'entrelaceurs ARP qui minimisent l'effet

des cycles de corrélation les plus courts sur le processus de décodage. Les contraintes identifiées sont résumées ci-après :

1. Minimiser le nombre de cycles de corrélation de longueurs les plus faibles,
2. Maximiser l'échange d'information entre les décodeurs sur les cycles de corrélation les plus courts,
3. Minimiser le nombre de fois où chaque couple de symboles participe à un cycle de corrélation court.

En appliquant ces contraintes, les entrelaceurs ARP offrant de bons gains de codage ont été obtenus pour les NB-TCs.

Poinçonnage des NB-TCs

Poinçonnage au niveau symbole

Les cycles de corrélation courts sont particulièrement affectés par le poinçonnage, car la suppression des symboles systématiques et/ou de parité participant à un cycle court de corrélation tend à réduire la diversité de codage. Ceci conduit naturellement à une corrélation plus élevée dans le processus de décodage. Afin d'atténuer cet effet préjudiciable à la correction des erreurs, nous avons conçu les modèles de poinçonnage de manière à maximiser le nombre de symboles non poinçonnés dans les cycles de corrélation courts.

Poinçonnage au niveau bit

Lorsque l'on considère des constellations d'ordres élevés, un nouveau degré de liberté peut être exploité. Ce degré de liberté est basé sur les différents niveaux de protection des bits de l'image binaire des symboles de la constellation. Profitant de cette différence dans les niveaux de protection, nous avons proposé une technique de poinçonnage. Cette technique offre de meilleures performances de correction que le poinçonnage classique des symboles lorsque l'on considère des constellations d'ordres élevés.

Spectre de distance et bornes de l'union

Dans cette section, nous avons proposé une méthode simplifiée pour estimer le spectre de distance euclidienne des NB-TCs lorsqu'ils sont associés à des modulations d'ordre élevé. Pour ce faire, nous avons identifié et caractérisé les modèles d'erreurs des NB-TCs dans la région des forts SNR, à savoir :

1. Erreurs dues aux faibles valeurs de *spread*,
2. Erreurs dues aux cycles de corrélation les plus courts.

Après avoir identifié les séquences sujettes aux erreurs, le spectre de distance euclidienne est calculé et les bornes de l'union sur les probabilités d'erreurs symboles et paquets sont évaluées pour les NB-TCs. Un exemple d'évaluation de ces bornes est illustré en Fig. 6, où C_1 et C_3 sont des NB-TCs construits à partir de deux NB-CC sur GF(64) définis au chapitre 2.

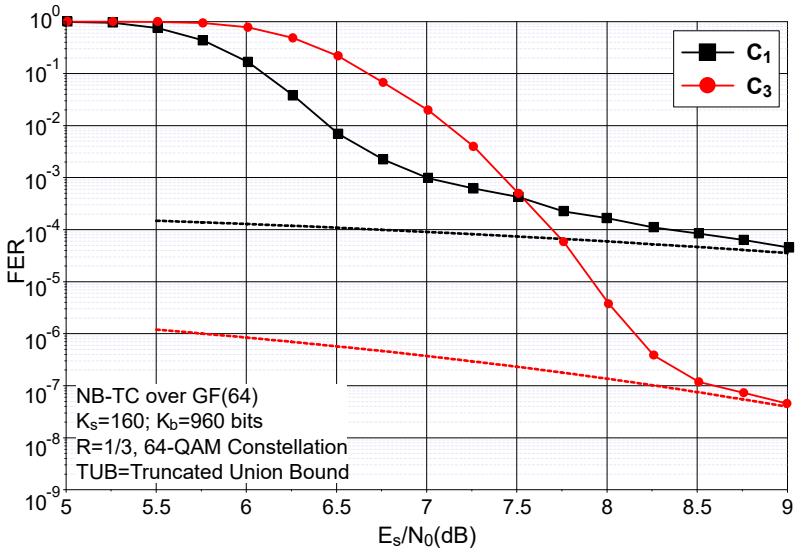

Figure 6: Évaluation sur canal gaussien du taux d'erreurs trames (FER) de deux NB-TCs associés à une constellation 64-QAM et comparaison avec les bornes de l'union tronquées (TUB) correspondantes.

Chapitre 4. Introduction d'une fonction de transformation pour améliorer les NB-TCs

Dans ce chapitre, nous proposons une modification de la structure des NB-TCs par l'addition d'une fonction de transformation des symboles avant l'encodage du message par l'un des encodeurs convolutifs. Cette modification permet d'augmenter le gain de codage des NB-TCs. Plusieurs comparaisons de performance des NB-TCs avec les codes de l'état de l'art sont également réalisées dans ce chapitre.

Structure modifiée des NB-TCs

La transformation peut être insérée avant l'encodage du message par le premier codeur RSC1 ou par le second codeur RSC2 comme le montre la Fig. 7. Dans le second cas, cette transformation peut être effectuée indifféremment avant ou après l'entrelacement.

La transformation considérée est une bijection sur l'ensemble des symboles du corps de Galois considéré: chaque symbole de GF(q) a une image unique dans GF(q) par cette transformation et tout élément de GF(q) a un antécédent unique dans GF(q).

Critères pour construire une transformation

Les séquences DC courtes avec de faibles CED provoquent souvent des erreurs dans l'un des décodeurs, mais le processus de décodage itératif permet de corriger la plupart de ces erreurs en utilisant les informations extrinsèques relatives aux symboles systématiques provenant de l'autre décodeur. Toutefois, le turbo-décodeur peut ne pas être en mesure de corriger ces erreurs lorsqu'un symbole transmis contribue à de courtes séquences DC avec des CEDs faibles dans les treillis de deux décodeurs, avec les mêmes symboles concurrents dans les deux paires de séquences DC. La participation d'un couple de symboles dans

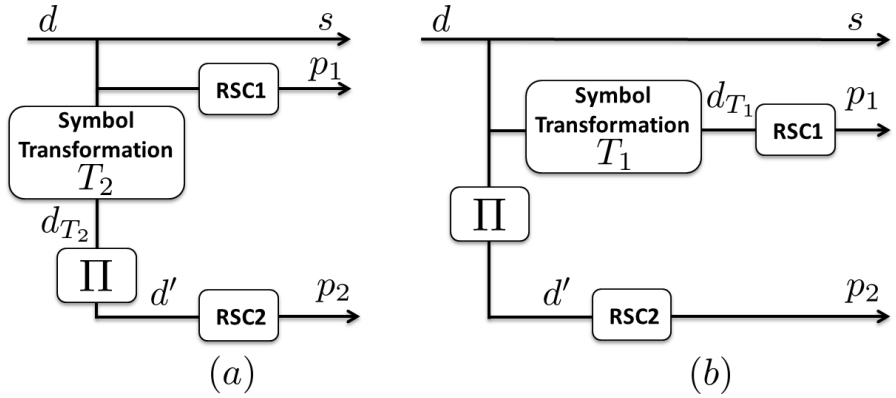

Figure 7: Structures modifiées du turbo-encodeur non binaire (a) avec transformation des symboles avant l'encodage du message par le second codeur RSC2 (b) avec transformation des symboles avant l'encodage du message par le premier codeur RSC1.

deux paires de séquences DC à faibles CED ne peut être évité même en utilisant un bon entrelaceur.

L'introduction de la transformation a pour objectif d'augmenter la distance euclidienne cumulée minimale entre deux séquences de symboles codés, indépendamment de la nature des codes élémentaires et de l'entrelaceur utilisés. Le critère d'optimisation lors de la construction de cette transformation consiste à maximiser la paramètre de dispersion, défini comme suit :

$$\delta(S_i, S_j) = d^2(S_i, S_j) + d^2(T(S_i), T(S_j)) \tag{2}$$

où S_i et S_j sont deux symboles de la constellation et $d(S_i, S_j)$ est la distance euclidienne entre S_i et S_j, et $d(T(S_i), T(S_j))$ la distance Euclidean entre leurs images après transformation.

Algorithme pratique de construction de la transformation dans le cas d'un code sur GF(64) associé à une 64-QAM

La maximisation du paramètre de dispersion par l'énumération de l'ensemble des transformations possibles devient irréalisable en pratique lorsque l'ordre de la constellation est élevé. Par conséquent, nous avons élaboré un algorithme sous-optimal de recherche de "bonnes" transformations permettant d'obtenir de grandes valeurs de $\delta(S_i, S_j)$ dans le cas d'un NB-TC sur GF(64) associé à une 64-QAM. Cet algorithme est basé sur plusieurs contraintes:

1. Transformer les symboles systématiques mappés à des points voisins de la constellation en des points non voisins avec une distance minimale garantie. L'objectif est d'éviter d'avoir des séquences DC à faible CED pour les deux codes convolutifs.

2. Transformer les symboles dans les coins et sur les bords de la constellation, naturellement mieux protégés par la constellation que les symboles intérieurs, en symboles intérieurs et *vice versa*. Ceci permet d'augmenter la protection, à l'entrée de l'autre codeur, des symboles les moins bien protégés.

En appliquant ces contraintes, nous avons trouvé des transformations permettant d'améliorer de manière significative les performances des NB-TC précédemment définis.

Algorithme pratique de construction de la transformation dans le cas d'un code sur GF(64) associé à une 4-QAM

Dans le cas d'un code sur GF(64) associé à une 4-QAM, l'algorithme de recherche est basé sur des contraintes différentes mais qui visent le même objectif vis-à-vis de la dispersion :

1. Transformer tous les couples de symboles sur GF(64) (S_i, S_j) ayant deux symboles 4-QAM constitutifs identiques en couples de symboles $(T(S_i), T(S_j))$ n'ayant aucun symbole 4-QAM en commun,

2. Transformer le nombre maximum de couples sur GF(64) (S_i, S_j) ayant un symbole 4-QAM constituant commun en couples GF(64) $(T(S_i), T(S_j))$ n'ayant aucun symbole 4-QAM commun.

Exemple de résultat

La Fig. 8 montre l'effet de l'application d'une telle transformation sur les performances de correction de NB-TCs sur GF(64) associés à une 64-QAM. On observe que l'introduction de la transformation permet d'abaisser le plancher d'erreur de manière significative et par conséquent d'améliorer le pouvoir de correction à faible taux d'erreur du turbocode.

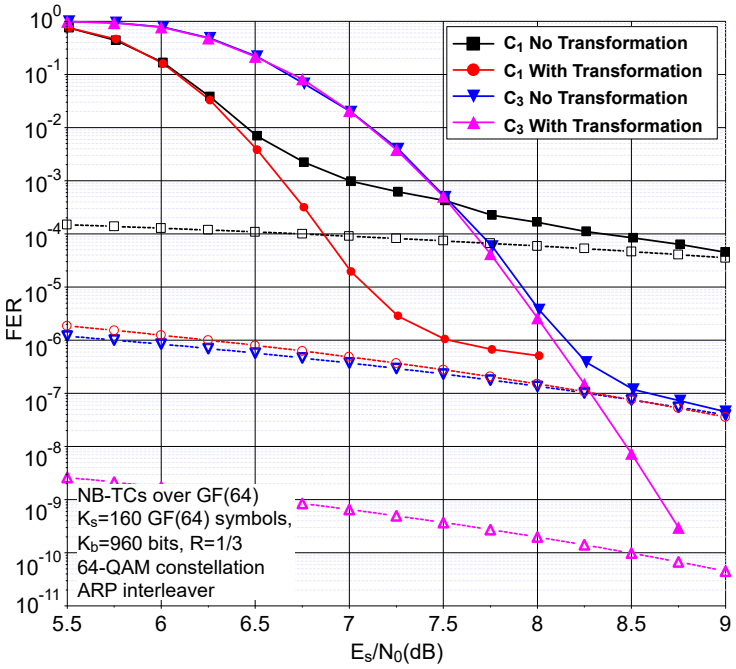

Figure 8: Comparaison du taux d'erreur trame avec l'entrelaceur ARP sur un canal gaussien et une modulation 64-QAM, $K_s = 160$ symboles sur GF(64) et $R = 1/3$ pour C_1 et C_3 dans GF(64) avec et sans transformation de symboles.

Exemples de performance et comparaison avec les codes de l'état de l'art

La dernière section de ce chapitre présente des courbes de performances des nouveaux codes NB-TCs et les compare avec les meilleurs codes binaires de l'état de l'art et avec

les codes NB-LDPC publiés dans la littérature. A titre d'illustration, les figures 9 et 10 montrent que les NB-TCs présentent des meilleurs performances que les codes LDPC et polaires adoptés dans la 5G.

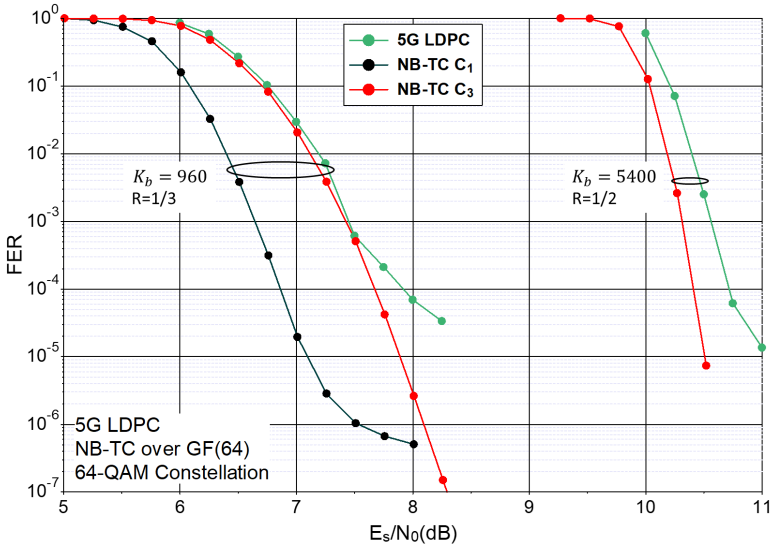

Figure 9: Comparaison du taux d'erreurs trames des NB-TCs proposés avec les codes LDPC adoptés dans le standard 5G.

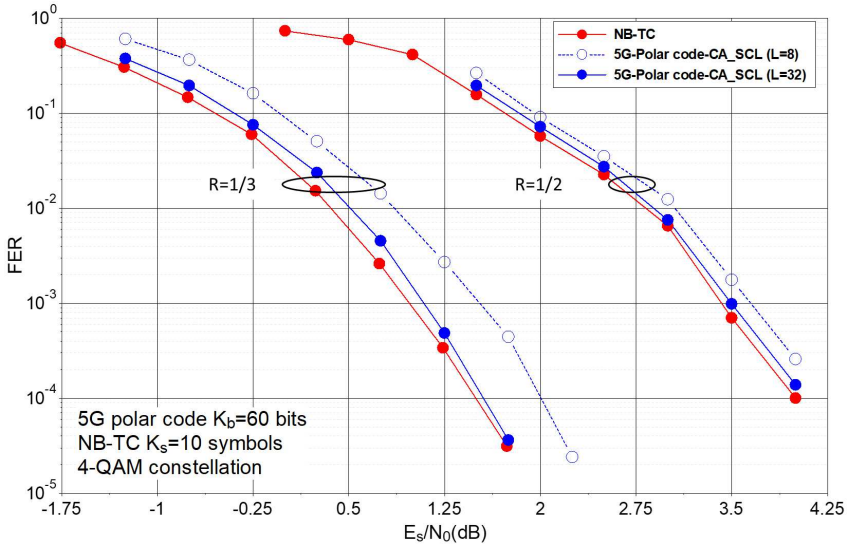

Figure 10: Comparaison du taux d'erreurs trames des NB-TCs proposés avec les codes polaires adoptés dans le standard 5G.

Chapitre 5. Decodeurs faible complexité pour les turbo codes non binaires

En raison de la structure des codes NB-TCs adoptés, leur complexité calculatoire de décodage ainsi que la mémoire nécessaire pour stocker l'information extrinsèque et les métriques d'états augmentent avec l'ordre du corps de Galois considéré. Pour de grandes valeurs de q, la complexité de décodage peut alors s'avérer trop élevée pour envisager des réalisations pratiques de décodeurs. L'objectif de ce chapitre est par conséquent d'analyser la complexité de décodage des algorithmes conventionnels puis de proposer un nouvel algorithme de décodage permettant de réduire la complexité du décodeur tout en préservant les bonnes performances des NB-TCs.

Analyse de la complexité de l'algorithme Min-Log-MAP

L'algorithme considéré est basé sur l'algorithme Min-Log-MAP [14] appliqué dans le domaine des symboles. Cet algorithme fait appel à des processus de calcul itératif des métriques d'état et des informations extrinsèques utilisant des opérations d'accumulation et de calcul de minima (min-somme). Sa complexité calculatoire varie comme q^2 pour la structure de code considérée. D'autre part, les q logarithmes de rapports de vraisemblance ou *log-likelihood ratios* (LLR) calculés par le démodulateur pour chaque symbole systématique et de parité reçu doivent être mémorisés à l'entrée du décodeur, soit un total de qK/R LLRs, si K est la longueur du bloc d'information et R le rendement de codage. Une mémoire est également nécessaire pour stocker les qK LLRs extrinsèques échangés pendant le processus de décodage itératif et, enfin, chaque décodeur Min-Log-MAP doit également stocker un certain nombre de metriques d'état pendant le processus de décodage. Dans le cas d'un décodage par fenêtres conventionnel, qW métriques doivent être mémorisées, si W est la longueur de la fenêtre.

Algorithme de décodage modifié

Pour réduire la complexité de stockage et la complexité calculatoire, nous proposons dans ce chapitre une variante de l'algorithme classique Min-Log-MAP :

- Les besoins en stockage sont réduits par l'application d'une technique de troncature pour les vecteurs de métriques de décodage qui n'entraîne pas de dégradation des performances,

- La complexité calculatoire est réduite par l'adoption d'une technique de *bubble check*. Le *bubble check* est un trieur 2D qui peut estimer les valeurs des métriques de décodage avec une complexité réduite.

Résultats

L'application de l'algorithme proposé permet de réduire la complexité tout en préservant la bonne performance des NB-TCs. Les figures 11 et 12 montrent quelques exemples de résultats. Lorsque l'on accepte une dégradation des performances de l'ordre de 0,15 dB, une réduction de la complexité calculatoire d'un facteur 5 peut être obtenue (courbes

magenta). Si la réduction de complexité requise est plus élevée (facteur 10 pour la configuration correspondant aux courbes bleues), la dégradation de performance de correction devient beaucoup plus importante, plus de 1 dB. Le compromis entre complexité et performance doit par conséquent être soigneusement choisi pour chaque application, afin de définir les meilleurs paramètres de l'algorithme de décodage.

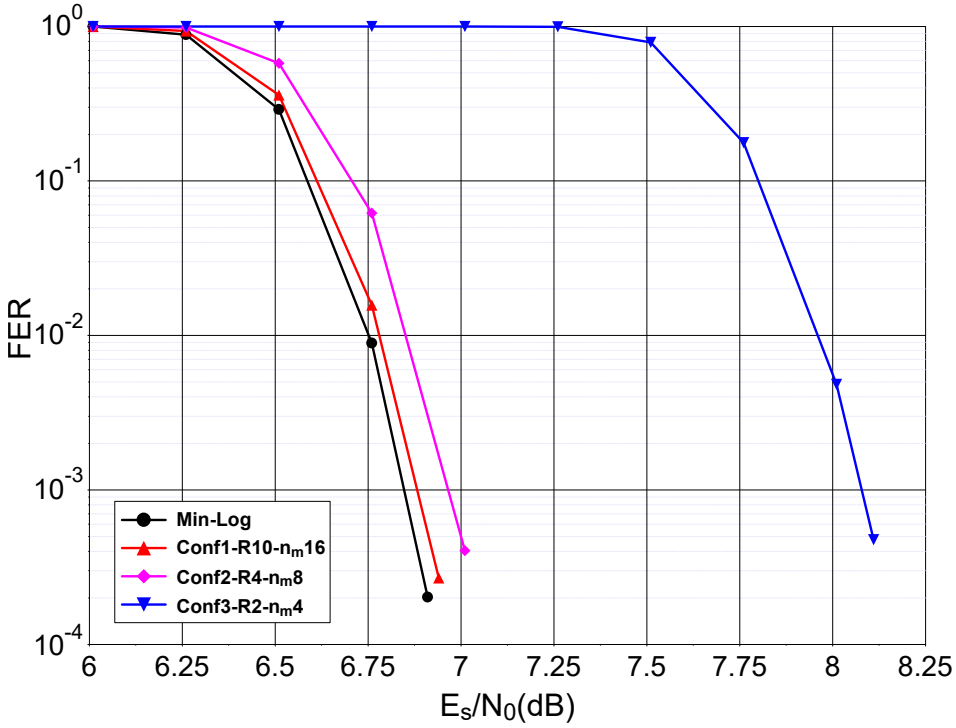

Figure 11: Comparaison des performances, en termes de FER, de 3 configurations de décodage à complexité reduite des NB-TCs. Algorithme Min-Log-MAP sur GF(64). Transmission avec une constellation 64-QAM sur canal gaussien. Paramètres du NB-CC : $(a_1,a_2,a_3) = (31,5,18)$.

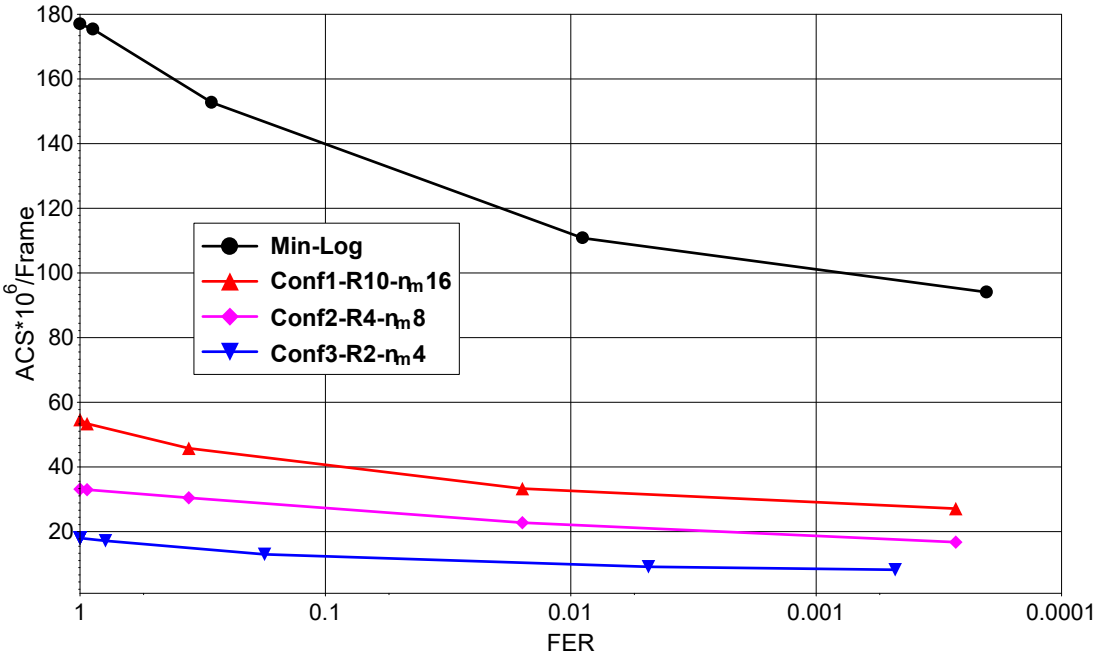

Figure 12: Comparaison de la complexité calculatoire des 3 configurations simulées en Fig. 11 en termes d'additions/comparaisons/sélections (ACS) en fonction du FER.

Conclusions et perspectives

Dans cette thèse, nous avons étudié une nouvelle famille de codes correcteurs d'erreurs basés sur des codes NB-TCs. Tous les éléments constitutifs sont abordés dans ce travail.

Tout d'abord, nous avons commencé par définir une nouvelle structure de codes NB-CCs. Le codeur constituant proposé est basé sur une nouvelle structure à un élément mémoire, qui permet d'augmenter de plus de 25% la distance euclidienne minimale de la structure conventionnelle d'accumulateur.

Ensuite, nous nous sommes concentrés sur un modèle de permutation qui peut offrir les meilleurs gains de codage pour le NB-TC résultant. Pour l'adaptation du rendement, nous avons étudié les différents types de poinçonnage qui peuvent être imaginés lors de l'encodage de symboles. Contraints par l'ordre de la constellation, nous avons pu proposer des schémas de poinçonnage au niveau bit qui offrent des meilleurs performances (gain de plus de 0,5 dB) que le poinçonnage au niveau symbole, pour des rendements de codage élevés. En outre, nous avons proposé une méthode pour estimer le spectre de distance tronqué, afin de calculer les bornes de l'union.

Afin d'améliorer les performances asymptotiques des NB-TCs, nous avons également inséré dans la structure de ces codes un nouveau bloc de transformation avant le second codeur constituant. Nous avons par ce biais réussi à abaisser le plancher d'erreur des NB-TCs de plus de deux décades, sans pour autant pénaliser leur performance aux taux d'erreurs élevés.

Enfin, un algorithme de décodage de faible complexité a été élaboré. Notre proposition a permis de réduire à la fois les besoins en mémorisation et la complexité calculatoire du décodeur.

La performance de correction d'erreurs des codes étudiés a été évaluée et comparée avec les codes de l'état de l'art pour différents scénarios. Nos NB-TCs offrent de meilleures performances comparativement aux meilleurs codes binaires et non-binaires de la littéra-

ture, pour différentes tailles de trame, rendements de codage et ordres de constellation.

De nombreuses perspectives peuvent être envisagées pour compléter ce travail. Par exemple, l'étude des NB-CCs peut être étendue à la conception des NB-CCs avec 2 éléments mémoires, qui peuvent permettre d'obtenir de meilleures capacités de correction des erreurs. De plus, les techniques de poinçonnage peuvent davantage être explorées en ciblant les systèmes de communication qui permettent la compatibilité de rendement. De plus, des optimisations plus fines du bloc de transformation peuvent également être recherchées. D'autre part, une implémentation matérielle est envisagée pour le décodeur proposé, dont il est démontré qu'il réduit considérablement la complexité calculatoire du décodeur non-binaire sans pour autant pénaliser l'excellente performance de ces codes.

Liste des publications

Les résultats de ces travaux ont donné lieu à plusieurs publications et un brevet:

Publications en conférences internationales

- R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah, "Design of low-complexity convolutional codes over GF(q)", in *IEEE Global Commun. Conf.: Communication Theory (Globecom2018 CT)*, Abu Dhabi, United Arab Emirates, Dec. 2018.

- R. Garzón Bohórquez, R. Klaimi, C. Abdel Nour and C. Douillard, "Mitigating correlation problems in turbo decoders", in 10$^{\text{th}}$ *International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018)*, Hong Kong, Hong Kong SAR, Dec. 2018.

- R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah, "Low-complexity decoders for non-binary turbo codes", in 10$^{\text{th}}$ *International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018)*, Hong Kong, Hong Kong SAR, Dec. 2018.

Publications en journal (soumissions)

- R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah "Union bound evaluation for non-binary turbo codes", submitted to *IEEE Trans. on Commun.*

- R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah "Enhancing performance of non-binary turbo codes via symbol transformation", to be submitted to *IEEE Trans. on Commun.*

Brevet

- R. Klaimi, C. Abdel Nour and C. Douillard "*Procédé de génération d'un signal mettant en oeuvre un turbo-codeur, dispositif et programme d'ordinateur correspondants*", Demande de brevet FR 1873615, Dec. 2018.

Introduction

Nowadays, communication systems have become an essential element in the development of the society. All human activities are becoming more and more dependent on the offered technologies. The evolution of the communication services involves: multimedia services, human-machine interaction, machine to machine communications etc. All these newly emerged services result in a significant increase in the number of the connected devices. Some studies [1] have shown that the number of connected devices will exceed the world's population by 2022, and will reach around 12.3 billion devices. In July 2018, the International Telecommunication Union launched a new focus group, FG NET-2030, to study the capabilities of networks for the year 2030 and beyond, which are expected to support novel forward-looking scenarios, such as holographic communications, autonomous vehicles, massive machine-type communications, tactile Internet, ... In the study of network 2030 architecture, new communication requirements are expected to emerge that are more sensitive to resource demands. Reliable and spectrally efficient communications are required to support the explosive bandwidth-intensive future applications beyond the limitation of near future 5G networks. Today's communication standards are not fully adapted to this tremendous growth. One of the limitations facing the evolution of the communication standards is the difficulty approaching the channel capacity limits drawn by Claude Shannon in [2], while communicating at high spectral efficiency. Different Forward Error Correcting (FEC) codes with good error correction capabilities were introduced in the literature, but only few were adopted in worldwide communication standards. Among them, Turbo Codes (TC) were introduced in 1993 by Claude Berrou et al. [3] and are known as the first implementable FEC codes that perform close to capacity limits. Low-Density Parity-Check (LDPC) codes, initially introduced by Robert G. Gallager in 1962 [15], are based on sparse parity-check matrices and were re-discovered in [4]. They are known to complement TCs for high-throughput applications. Recently polar codes, proposed in 2009 by Erdal Arıkan [5] and based on the channel polarization principle, were adopted in the 5G new radio interface for the control channel. They are also able to achieve or approach channel capacity limits for different transmission scenarios.

Moreover, with the increase in the number of communicating applications that share the same radio spectrum, the spectral resources are becoming scarce. To remedy the problem of limited frequency spectrum resources, high spectrally efficient solutions should be considered. These often resort to the use of high-order modulations. As stated in [16], when using high-order modulations, Coded Modulations (CM) can achieve higher spectral efficiencies for the same signal-to-noise ratio than the Bit-Interleaved Coded-Modulation (BICM). This is particularly true for practical operating points at low and medium coding rates (below $R = 2/3$). The performance gap between BICM and CM can be efficiently bridged with shaping techniques, especially probabilistic shaping. Such techniques prove to work well for optical communications [17] but additional work still

needs to be performed to improve their suitability for wireless communications especially with mobility. Therefore, this motivates the study of the association of Non-Binary (NB) codes and high-order modulations with the same alphabet order. Different NB codes already exist in the literature: amongst the oldest, Reed-Solomon codes [18] are certainly the most popular, and non-binary LDPC (NB-LDPC), initially proposed in [19], have been the subject of multiple studies in the recent years, e.g. [20, 21, 22]. NB-LDPC codes are shown to outperform their binary counterparts, especially when mapped to high-order modulations.

Surprisingly, despite their earlier discovery, only less than a handful of papers can be found regarding non-binary convolutional codes (NB-CCs) or turbo codes (NB-TCs). The structure of TCs, which is based on the parallel concatenation of two or more convolutional codes, offers an excellent tradeoff between performance and complexity when performing at low coding rates. Lately with recent advances in interleaver designs, TCs are shown to reach low error rates even when high coding rates are considered [13]. Therefore, NB-TCs can provide a promising encoding template in the context of FEC codes designed to be mapped to high-order modulations.

The aim of this thesis is to start to provide the framework and tools for the design of NB-TCs. To this end, we address each of the building blocks one by one. A structure of recursive systematic NB-CCs is proposed first. Then, interleaving and puncturing techniques are adapted to the NB domain. Based on the work presented in [13], Almost Regular Permutation (ARP) interleavers are shown to be well suited for NB-TCs, while two puncturing methods are identified and assessed. We also propose to introduce an additional block in the NB turbo encoder structure, which is shown to enhance the error correction capability of the NB-TC in the error floor, without entailing any loss in the waterfall region. Finally, a low-complexity decoding algorithm is proposed for these codes in order to pave the way towards future hardware implementations. On another note, to design efficient codes, tools such as union bounds are used to predict asymptotic performance. However, the extension of these tools to NB codes is far from being straightforward. In this thesis, we also propose a new way to identify error events and patterns that play a dominant role in the computation of the adapted bounds for such codes.

Thesis outline

Chapter 1 first explains the basic notions of channel coding as well as the channel capacity limits and stresses the necessity of studying new error correction codes. Then, we state the motivations for studying NB error correction codes and a state-of-the-art review of the design and performance of NB-LDPC codes is presented. Next, we explain the motivations for studying TCs in the NB domain and we introduce the necessary technical background about TCs.

In Chapter 2, a design of low-complex non-binary convolutional codes is proposed. The chapter first provides a summary of the calculation techniques in the Galois fields, and presents state-of-the-art structures of NB-CCs and NB-TCs. Then, we describe our proposal for a new encoding template for NB-CCs based on an updated structure of an accumulator. Next, a method to calculate the distance spectrum of the NB-CCs is proposed: sequences creating low cumulated Euclidean distances are identified, and a simple algorithm is presented that enumerates the lowest two Euclidean distances with their

corresponding multiplicities. Then, we discuss the impact of the constellation mapping on the distance spectrum of the code. After having presented the error correction performance of several examples of NB-CCs designed over GF(4), GF(16) and GF(64), two additional techniques are proposed in order to enhance the performance of these codes when mapped to modulations of lower order than the Galois field of the code.

The third chapter is devoted to the design of NB-TCs, by addressing the different blocks of the TC structure other than the constituent NB-CCs. This chapter begins by studying the effect of different interleaving techniques on NB-TCs and a method is proposed, that targets the mitigation of the correlation problems in the short correlation cycles when short frame sizes are considered. Next, two puncturing techniques are proposed. The last section of this chapter is dedicated to the practical computation of the union bounds for NB-TCs. A method is proposed to estimate a truncated cumulated Euclidean distance spectrum of NB-TCs by identifying the structure of competing sequences in the code trellis with low cumulated Euclidean distances.

In Chapter 4, the insertion of a symbol transformation before one of the NB component encoders is investigated. The design criterion to define the transformation connections is identified, and we derived an algorithm to define good transformation connections when a q-QAM constellation is used with NB-TCs over GF(q) as well as when a 4-QAM constellation is considered. Next, different application examples are investigated to assess the performance of the resulting NB-TCs, and comparisons are conducted with state-of-the-art binary and NB codes.

Chapter 5 studies a new low-complexity decoding algorithm for the proposed NB-TCs. The first section presents state-of-the-art decoding techniques for NB codes. Next, the main steps of the classical scaled Min-Log-MAP algorithm for NB-TCs are summarized with their corresponding decoding complexity and storage requirements. A simplified algorithm to decode NB-TCs is then presented, which lowers the storage requirements as well as the computational complexity. In particular, the number of performed ACS operations is reduced using a technique inspired from the bubble check sorter previously proposed for NB-LDPC codes. Different application examples are presented and our proposal is shown to offer high complexity reduction, at the price of a limited performance penalty still keeping most of the superior performance of the NB-TCs. This decoding algorithm can be seen as the first step towards possible hardware implementations of the proposed structure of NB-TCs.

The last chapter concludes this thesis and suggests future works that could be conducted in this research domain.

Contributions of the thesis

This work contains original contributions that are proposed in order to define the structure of NB-TCs. Below is a summary of the main contributions of this thesis:

- The main structure of NB-CCs that are used as component encoders for NB-TCs has been addressed in Chapter 2. We proposed a new low complex encoding template, while targeting codes that offer better performance than binary CCs. The main contributions of this chapter are:
  - proposing a new memory-1 convolutional code structure with increased Euclidean distance,

  - defining a constellation downsizing process, that further improves the Euclidean distance of the underlying NB-CC while reducing the receiver-side complexity,
  - the procedure to optimize the code when associated to high-order and binary modulation types.

- In chapters 3 and 4 the following contributions can be listed:
  - the criteria to design efficient ARP interleavers able to mitigate most of the effects of short correlation cycles (joint contribution with Dr. Ronald Garzón Bohórquez),
  - two puncturing types are proposed, which are shown to offer good performance for NB-TCs when targeting high coding rates,
  - a method is proposed to estimate the distance spectrum of these codes, based on the enumeration of the most probable sequences that can result in low cumulated Euclidean distances,
  - union bounds are adapted to the context of the NB-TCs, and are found to represent an efficient design tool,
  - a new block is added to structure of NB-TCs, based on a bijective transformation, increasing the coding gain of these codes by avoiding the repetition of error-prone competing sequences at both decoders.

- A low-complexity decoder is designed in Chapter 5 of this thesis:
  - a vector truncation is applied to limit the storage requirements of the Min-Log-MAP decoder,
  - a new decoding method is proposed, based on a newly derived bubble sorter, reducing drastically the number of the performed Add Compare Select (ACS) operations.

Chapter 1

Channel coding and non-binary codes

New communication applications are progressively coming to light, imposing new demands that should be satisfied by the communication systems. To this end, different channel coding techniques have been introduced to satisfy stringent error rate constraints as well as high data rates. In this context, new error correction codes need to be introduced, and non-binary coding schemes have been recently proposed as a solution to some drawbacks of the existing coding schemes. In this chapter, an overview on the channel coding concepts is given as well as the motivations to study non-binary codes. Section 1.1 begins by describing the basic notions of channel coding, as well as the channel capacity limits in Section 1.1.1. Then, Section 1.1.2 explains the necessity to study new error correction codes. In Section 1.2, an overview of existing non-binary codes is given. First, the definition of non-binary codes is provided in Section 1.2.1. Then, the construction of non-binary LDPC codes and their comparison with binary LDPC codes are conducted in Section 1.2.2. In the following, the motivations for the study of non-binary turbo codes are explained in Section 1.2.3. The perquisites for understanding and designing turbo codes are presented next. Finally, Section 1.4 concludes the chapter.

1.1 Generalities on channel coding

In 1948, Claude Shannon introduced the channel coding theory [2] by demonstrating that a reliable communication can be achieved even when transmitting information through a noisy channel. Channel coding is defined as the tool enabling the correction of transmitted packets through an error-prone communication channel. Since then, many error correction codes were introduced, beginning by binary Golay codes in 1949 [23], Hamming codes in 1950 [24], convolutional codes in 1955 [25] and BCH codes in 1959 [26]. In 1993 a new era of error correction codes was reached, where turbo codes (TCs) [3] were defined by Claude Berrou et al. as the parallel concatenation of several (in general two) constituent encoders. Turbo decoding is based on iterative decoding, where component decoders exchange extrinsic information, thus enabling to closely approach the theoretical coding limits set by Claude Shannon in [27]. The introduction of turbo codes has drawn the attention of the channel coding research community to another interesting fact: they were the first error correcting codes able to approach the Shannon limit with a decoding complexity sufficiently low to allow their implementation in practical communication systems. In 1997, David Mackay has re-introduced Low Density Parity Check codes (LDPC) [4] inspired from an already existing structure which was devised by Robert Gallager in 1962 [15]. In 2009, a new family of error correcting codes is defined by Erdal Arıkan [5]. Based on the channel polarization principle, these codes were named *polar codes*, and they were shown to achieve the channel capacity limits when high frame sizes are used.

1.1.1 Channel Capacity

Before 1949, in the pre-coding era the only solution to lower the error probability of a transmission over a noisy channel was the repetition of the information. In 1949, Claude Shannon drew the theoretical limits for error correction codes [27], where he demonstrated that, by coding the information, error-free communications can be achieved with rates lower than the channel capacity. Considering a Gaussian channel, it is demonstrated that the maximum capacity C can be expressed as:

$$C = \log_2(1 + \mathrm{SNR}) \tag{1.1}$$

where SNR is the received signal to noise ratio. C expresses the maximum number of bits/second/Hz that can be transmitted through the channel with an arbitrarily low probability of error (~without error). To approach this theoretical capacity, one should investigate the design of error correction codes enabling high error correction capabilities.

1.1.2 Motivation for studying new error correcting codes

It is known that TCs, LDPC and polar codes approach capacity limits in different scenarios. These codes have paved the way for efficient hardware implementations and have already been adopted in different communication standards [28, 29]. However, with the advent of new communication standards, new demands have surfaced. For applications requiring low latency and high throughput, codes achieving very low error floors are of high importance for avoiding packet re-transmission. Additionally, robust error correction capabilities are required for very short frame sizes in the context of Ultra Reliable Low Latency Communications (URLLC) and the Internet of Things (IoT) scenarios. Moreover,

longer device battery life and low latency applications such as autonomous vehicle and remote surgical procedures are also to be supported. These constraints call for contradicting features in the sense that improved performance often calls for additional complexity (and hence consumption) and/or latency. Therefore, none of the already defined codes can simultaneously satisfy all these new demands, and the introduction of a new error correction code, trying to bring us one step further into fulfilling all these emerging demands, seems of high importance.

1.2 Capacity-approaching non-binary codes

1.2.1 Motivation for the study of non-binary codes

In the literature, several non-binary (NB) codes have already been defined. Reed-Solomon codes are defined in [18] as the extension of BCH codes in the NB domain. Davey and MacKay extended the LDPC codes to the NB domain in [19]. These codes have shown a real potential in outperforming their binary counterparts. Several issues motivate the study of error correction codes in the NB domain:

1. When small frame sizes are considered, capacity-approaching binary codes such as LDPC codes are shown to have a drop in performance compared to high frame sizes, where it is shown that they do not approach closely the channel capacity limits [6]. This is mainly due to the high correlation experienced in the iterative decoding process [30]. When the code operates on an alphabet A of cardinality $q > 2$, a high number of symbols represents the information data, which reduces the correlation effects in the iterative decoding process [31] and helps resisting burst errors [32]. In conclusion, when NB codes are introduced, lower correlation is experienced.

2. High order modulations can be used to increase spectral efficiency, thus enabling high throughput transmissions. However, when binary codes are mapped to high order modulations, using the well-known bit-interleaved coded modulation (BICM) scheme [33, 16, 34], the resulting error correction performance can be far from the Shannon limit. As discussed later in Section 1.2.3, NB codes can display better theoretical performance than binary codes when associated with high-order modulations. This is one of our motivations to design NB codes.

3. More degrees of freedom are offered when designing NB codes, compared to binary ones. First, in the case of NB turbo codes, for a given memory length, the search space for good convolutional codes is larger in GF(q) than in GF(2). If properly investigated, they can lead to more robust codes. Moreover, the non-binary nature of the code allows the possibility of introducing a symbol transformation applied on the input of the second component code of the turbo coding structure. This is investigated in our work in Chapter 4.

1.2.2 Review of existing non-binary LDPC codes and comparison with their binary counterparts

In 1962, Robert Gallager developed a new structure of error correction codes, based on the use of LDPC [15] matrices. However, this work was set aside for a while, since its decoding

algorithm was too complex to be implemented at that time. With the introduction of TC by Berrou in 1993 [3], the first practical error correcting codes and decoders able to perform very close to channel capacity were proposed. In the light of the iterative decoding process of TCs, several researchers got interested in re-discovering LDPC codes in order to decode them with such decoding algorithms, mainly David Mackay in [4] and Michael Sipser in [35].

LDPC codes are block codes having sparse parity-check matrices. Usually, block codes are designed for short frame sizes, since their decoding algorithms are based on Maximum Likelihood (ML) decoding, whose complexity increases exponentially with the codeword size. On the contrary, due to the sparseness of their parity-check matrix, LDPC codes can be decoded using an iterative process. This helps reduce the decoding complexity, making decoders practically implementable, even for large block sizes. They have been the main competitors of TCs for many years. The iterative decoding for the LDPC codes is based on its graphical representation, known as bipartite graph or Tanner graph [36].

As an extension to the NB domain, NB-LDPC codes were first introduced in [19]. These codes have shown promising error correction capabilities, alongside with a reasonable hardware complexity for nowadays, making them of high importance to the coding research community.

1.2.2.1 Construction of NB-LDPC codes over GF(q)

NB-LDPC codes are defined over the Galois Field GF(q), $q > 2$, instead of GF(2) [19]. This means that the information frame, the codeword and the coefficients in the generator matrix and parity-check matrix are elements of GF(q). All the mathematical operations for the encoding and decoding processes are also performed in GF(q). Note that the definitions of the arithmetic operations in the GF(q) will be provided in Chapter 2, Section 2.1. An example of parity-check matrix for a code defined over GF(q) is given by (1.2), where $h_{ij} \in \{\text{GF}(q)\backslash\{0\}\}$.

$$H = \begin{pmatrix} 0 & h_{01} & 0 & h_{03} & 0 & h_{05} \\ 0 & 0 & h_{12} & h_{13} & h_{14} & 0 \\ h_{20} & 0 & h_{22} & 0 & 0 & h_{25} \\ h_{30} & h_{31} & 0 & 0 & h_{34} & 0 \end{pmatrix} \tag{1.2}$$

The parity check matrix shown in (1.2) presents an example of a NB-LDPC construction. All the codewords c, such that $c = [c_0; c_1; c_2; c_3; c_4; c_5]$ with $c_i \in \text{GF}(q)$, should verify $cH^T = 0$:

$$\begin{aligned} h_{01}c_1 + h_{03}c_3 + h_{05}c_5 &= 0 \\ h_{12}c_2 + h_{13}c_3 + h_{14}c_4 &= 0 \\ h_{20}c_0 + h_{22}c_2 + h_{25}c_5 &= 0 \\ h_{30}c_0 + h_{31}c_1 + h_{34}c_4 &= 0 \end{aligned} \tag{1.3}$$

The corresponding Tanner graph is depicted in Fig. 1.1. Unlike binary codes, permutation nodes are added to the graph edges, to perform the multiplication of the information exchanged between variable and check nodes by non-zero coefficients h_{ij}. Note that this example is given for illustration purposes; in practice, LDPC matrices are more sparse.

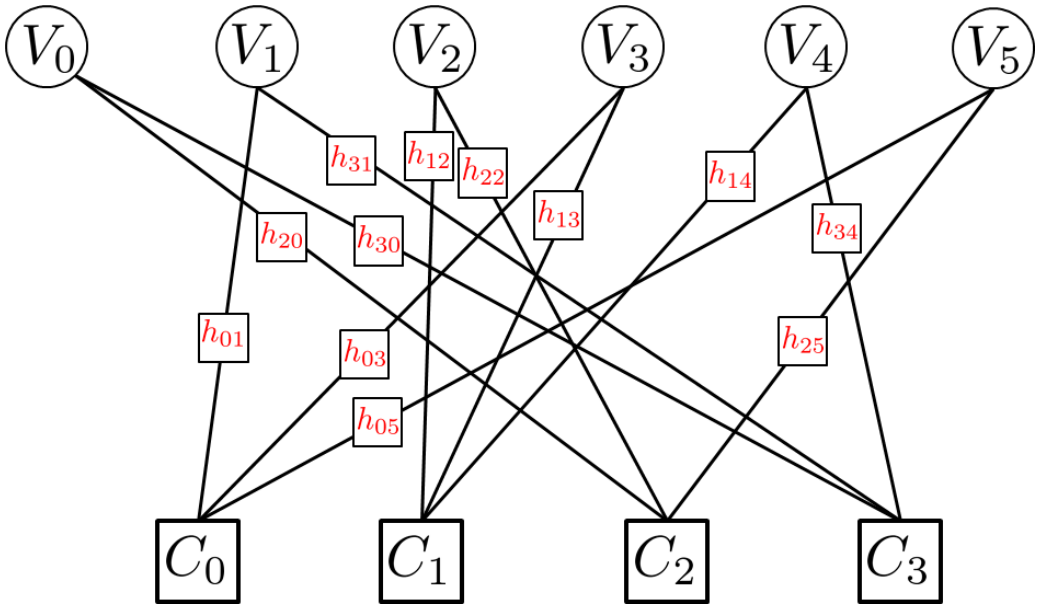

Figure 1.1: Tanner graph of a NB code represented by its parity-check matrix (1.2).

The authors in [19] reported a significant gain in performance of NB-LDPC codes when compared with binary LDPC codes. In [20], Mackay has shown different optimized coefficients for parity-check matrices over GF(q).

A new method to optimize the NB coefficients is proposed in [21, 22]. Based on the binary image of the NB codes, the authors proposed to investigate the algebraic properties of the NB-LDPC Tanner graph in order to define the best coefficients. A regular NB-LDPC is considered with (d_v, d_c, N) representing the number of non-zero elements in the columns of H, the number of non-zero elements in the rows of H and the code length respectively.

First, it is demonstrated that selecting good row coefficients can improve the waterfall region of the code. In [19, 20], the row coefficients are identified using Monte-Carlo simulations, where the sets of possible d_c coefficients in each row are identified in order to maximize the marginal entropy of the message after a given number of iterations. From the identified sets of d_c coefficients, the authors have proposed to form the matrix H by permuting and multiplying the identified coefficients by a constant.

In [21, 22], a new method is proposed in order to define these coefficients, where the authors have reported that both methods converge to the same results, with the advantage of a reduced computational complexity in the new method. The proposed method is based on the acceleration of the search procedure for the rows elements. The maximization of the minimum distance $d_{\min}$ of the associated binary code is seen as the goal of this method. H_i corresponds to the equivalent binary parity-check matrix, and $d_{\min}$ is equivalent to the minimum number of dependent columns in H_i. The minimum possible value of $d_{\min}$ is 2, and the goal is to achieve the highest value of $d_{\min}$, with the smallest multiplicity. To simplify the search procedure, each element of the possible set can be written as : $1 \dots \alpha^i \dots \alpha^j \dots \alpha^k$ corresponding to the d_c elements of the row. Moreover, a set with d_c elements can be derived from the primitive set of $d_c - 1$ elements by adding a new field, leading to $1 \dots \alpha^i \dots \alpha^j \dots \alpha^k \alpha^l$. Once the primitive set is computed for a row, some d_c-uplets can be deduced for other rows by a constant multiplication.

The second method for optimization proposed in [21, 22] consists in canceling short

cycles and in mitigating stopping sets in order to enhance the minimum distance, which results in avoiding high error floors. The steps of this algorithm can be summarized as follows:

1. The coefficients of the rows in the parity check matrix H are chosen randomly from the set $\mathcal{R}$, which denotes the set of optimized rows.

2. This step consists on the cancellation of the initial cycles with length l, such that $g \leq l \leq l_M$, with g denotes the girth of the parity-check matrix. The following procedure is iterated for $l \geq g$:

   (a) A random row m is selected from H.

   (b) A set of random permutations is selected from $\mathcal{R}$.

   (c) The permutation that maximizes the number of canceled cycles of length l is selected, noting that shorter cycles are already canceled.

   (d) If all the cycles at l are already canceled, l is incremented and the algorithm re-iterates. Else, the algorithm continues to cancel cycles corresponding to l.

3. In this step, cycles with length $l \geq l_M$ are intended to be canceled following the same procedure described above. When the cycle cancellation is performed on all the rows, the permutations are chosen in order to maximize the minimum distance over all the stopping sets.

4. When no more cycle cancellation is possible, the algorithm ends. It is shown that the minimum distance is improved, which lowers the error floor as well.

1.2.2.2 Performance comparison of binary and non-binary LDPC codes

As shown in [37], the proposed NB-LDPC codes outperform their binary counterparts when combined with a binary modulation, for different coding rates. Fig. 1.2 shows the comparison of the designed NB-LDPC codes over GF(256) in [37] with protograph irregular binary LDPC codes [38]. Obviously, it is shown that NB-LDPC codes can outperform the binary codes with more than 0.25 dB in all the proposed scenarios.

Moreover, different studies were conducted in order to design NB-LDPC codes associated with high order modulations [39, 40, 41]. NB-LDPC codes were also shown to outperform their binary counterparts when mapped to high-oder modulations. The results in [41] show interesting performance gains for non-binary codes over binary codes for high coding rates and with the use of high order modulation.

In Fig. 1.3, performance comparison between a NB-LDPC code over GF(256) with a binary LDPC is conducted over a rotated 256-QAM constellation. Obviously, NB-LDPC is shown to outperform the binary codes by almost 0.4 dB in the context of the high order modulation.

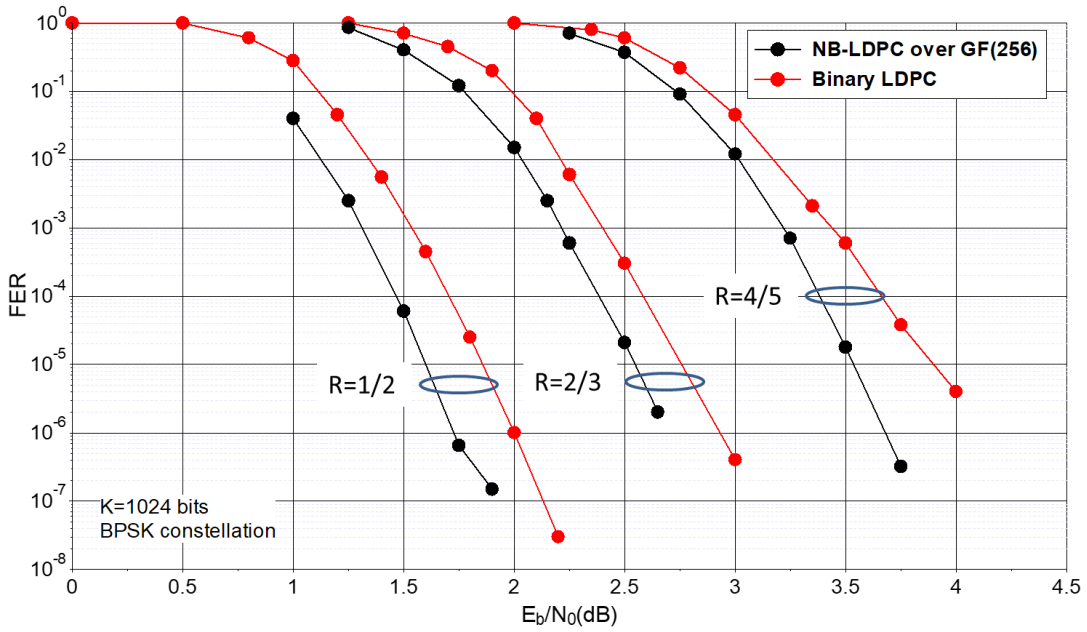

Figure 1.2: Performance comparison in terms of FER between NB-LDPC codes over GF(256) [37] and binary LDPC codes [38]. AWGN channel is used, $K = 1024$ bits, and different coding rates are studied.

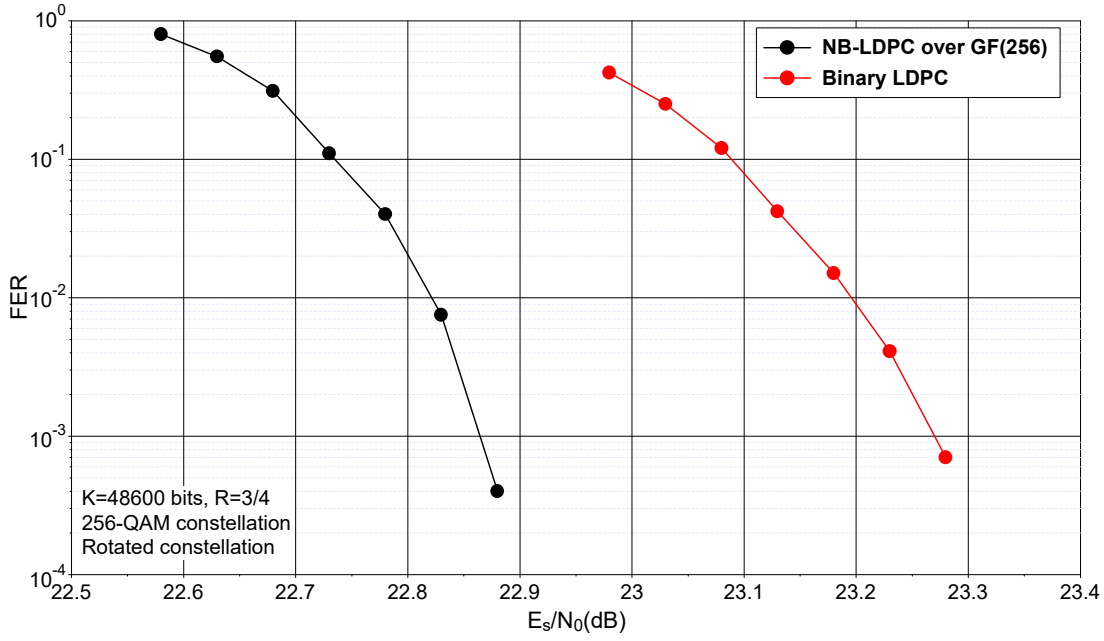

Figure 1.3: Performance comparison in terms of FER between NB-LDPC codes over GF(256) and binary LDPC codes as shown in [41]. A fast flat Rayleigh fading channel is used, $K = 48600$ bits, and $R = 3/4$. Rotated constellation is used, with an angle of 31.7° for the NB-LDPC, and 3.58° for the binary LDPC.

1.2.3 Motivation for the study of non-binary turbo codes

1.2.3.1 Codes mapped to high-order modulations: an information theoretic approach

As already introduced in the previous section, when high order constellations are considered, the question arises as to whether NB codes outperform binary codes or not. To discuss this issue, a comparison of the capacities of different coded modulation types is performed. The general block diagram of the communication channel under consideration is plotted in Fig. 1.4. E and D stand for the encoder and decoder blocks respectively, the channel interleaver is represented by Π, and the modulation is represented by the function μ.

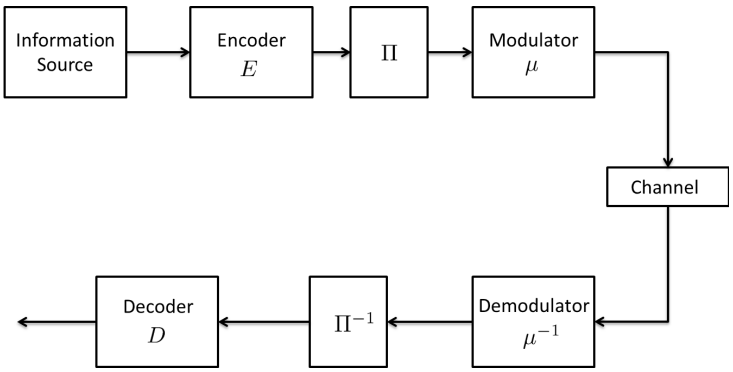

Figure 1.4: Block diagram of a transmission with Coded Modulation (CM) or Bit Interleaved Coded Modulation (BICM).

Different types of coded modulation can adapt to this diagram. Bit Interleaved Coded Modulation (BICM) was first introduced by Zehavi in [33], and was then analyzed by Caire in [16]. When BICM is considered, the encoder is designed in the binary domain, and the channel interleaver Π from Fig. 1.4 is a bitwise interleaver. On the contrary, Coded Modulation (CM) is the concatenation of an encoder defined over an alphabet $\mathcal{A}$ of size q, with a modulation defined over the same size q. In this case, the channel interleaver Π, when it is present, is also defined in the symbol domain, operating on the same non-binary alphabet $\mathcal{A}$.

As stated in [2], the channel capacity can be computed as the maximum value of the average Mutual Information (MI) between the channel input and output. The MI between two events r and x is defined as the amount of information related to event r, given the occurrence of an event x [42]:

$$I(x;r) = \log_2\left[\frac{p(r|x)}{p(r)}\right] \tag{1.4}$$

If X and R represent the input and output random variables of the considered channel, the channel capacity is defined as:

$$C = \max\{I(X;R)\} \tag{1.5}$$

where $I(X;R)$ is the average MI (AMI) between X and R. $I(X;R) = E[I(x;r)]$, where x and r are realizations of X and R, respectively. As developed in [34], the corresponding

capacity for a CM is calculated as follows:

$$C^{CM} = n - E\left[\log_2 \frac{\sum_{a\in\mathcal{A}} p(r|a)}{p(r|x)}\right] \tag{1.6}$$

where $\mathcal{A}$ represents the used alphabet and $n = \log_2|\mathcal{A}|$ the number of bits per constellation symbol. The corresponding capacity for a BICM is calculated as follows [16]:

$$C^{BICM} = n - \sum_{k=0}^{n-1} E\left[\log_2 \frac{\sum_{a\in\mathcal{A}} p(r|a)}{\sum_{a\in\mathcal{A}_{k,b}} p(r|a)}\right] \tag{1.7}$$

where $\mathcal{A}_{k,b}$ is the subset of $\mathcal{A}$ where the bit at position k is equal to b, $b \in \{0,1\}$.

1.2.3.2 Comparison of CM and BICM capacities for high-order modulations

Based on Equations (1.6) and (1.7), we conducted a comparison between the capacities of different Quadrature Amplitude Modulation (QAM) orders in the context of BICM and CM. Figures 1.5, 1.6, 1.7 and 1.8 show the CM and BICM capacity curves for QPSK, 16-QAM, 64-QAM and 256-QAM modulations, respectively. We can observe that, except for the QPSK case where channel capacity is identical for CM and BICM, using CM always yields better capacity than BICM. The capacity gain of CM over BICM is shown in Fig. 1.9, as a function of the spectral efficiency, for the four constellation types. It clearly shows that non-binary codes can be more advantageous when high modulation orders and low rates are considered. No gains are observed when comparing the capacities in the case of a QPSK constellation. One can conclude that, when using a QPSK constellation, binary and non-binary codes can have the same performance. This gain is more pronounced for higher modulation orders and at low spectral efficiencies. For example, when a 16-QAM constellation is considered, a gain in the order of 0.5 dB can be observed when using CM over BICM, for a spectral efficiency $C = 1$ bit/channel use. In this context, a non-binary code can outperform a binary code with 0.5 dB if the two systems are designed to approach their theoretical limits in the same manner. A gain of more than 0.9 dB can be observed for a 64-QAM when 1 bit per channel use is considered, and a gain in the order of 1 dB is observed for a 256-QAM constellation when less than 2 bits per channel use are considered.

To summarize, NB codes can bring significant coding gains over binary codes, provided that the order of the Galois Field is high enough and that low coding rates are considered. For high rates and low order modulations, no capacity gain is to be expected for NB codes over binary codes.

Previous comparisons between TCs and LDPC codes [43] showed that TCs are better suited than LDPC codes for low coding rate schemes. Well designed TCs can offer good performance for low coding rates, and can require low decoding complexity since their widely-used decoding algorithm operates on the information frame size and not on the codeword size. In this context, designing NB-TCs appears to be a promising study direction that is expected to compete advantageously with the existing NB-LDPC codes of the literature. NB-TCs have not been much investigated so far; in particular,

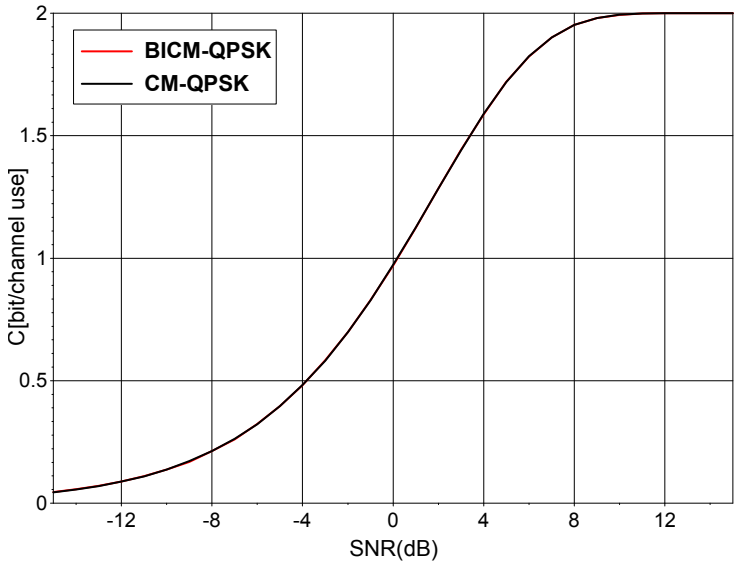

Figure 1.5: BICM and CM capacities versus SNR over AWGN channel for a QPSK constellation.

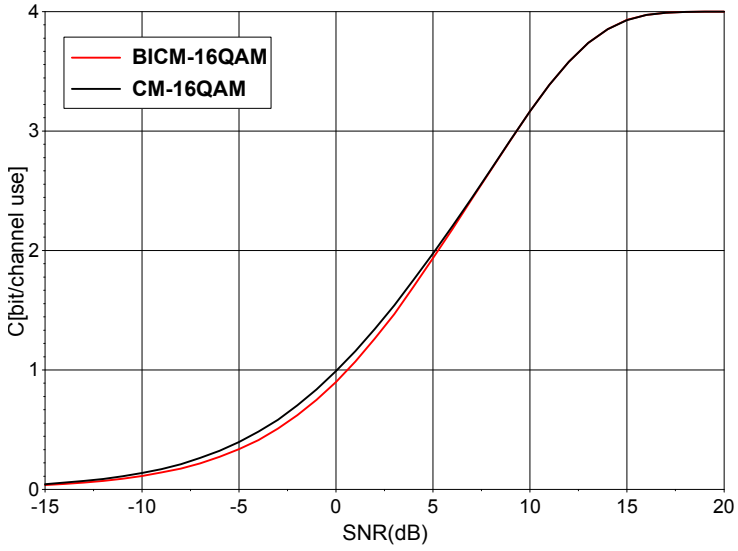

Figure 1.6: BICM and CM capacities versus SNR over AWGN channel for a 16-QAM constellation.

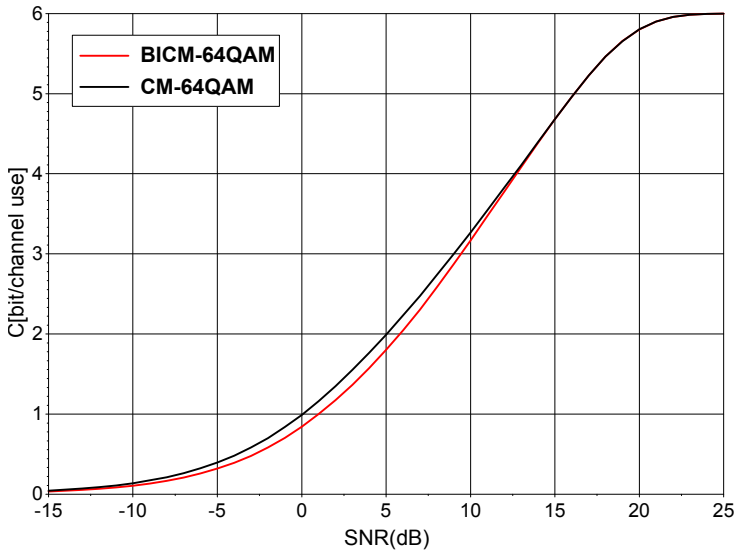

Figure 1.7: BICM and CM capacities versus SNR over AWGN channel for a 64-QAM constellation.

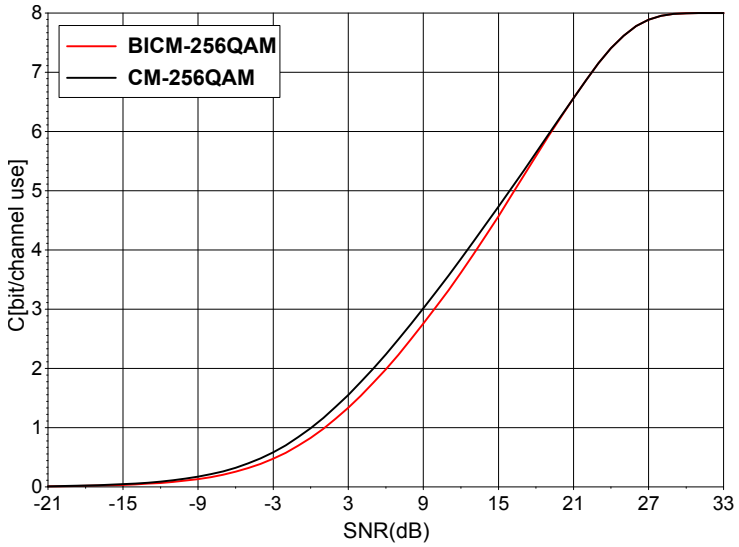

Figure 1.8: BICM and CM capacities versus SNR over AWGN channel for a 256-QAM constellation.

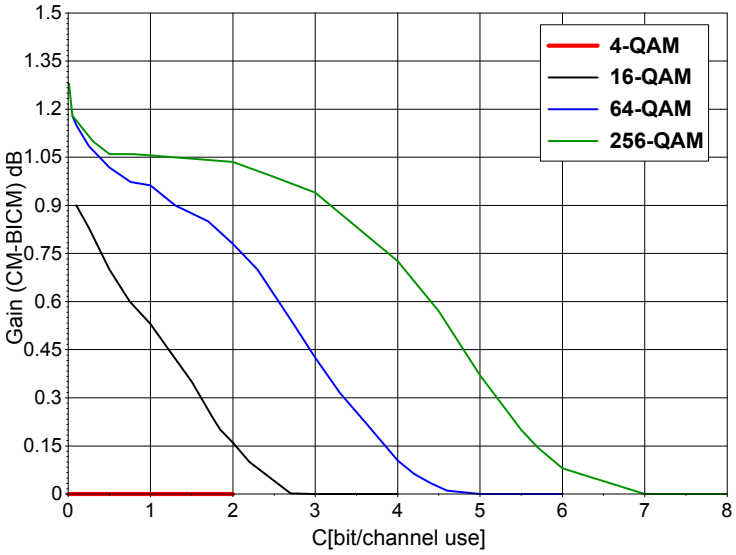

Figure 1.9: CM capacity gain over BICM for different constellation types, in function of the spectral efficiency.

a very few number of papers have studied the search for good NB component codes. In [44, 45, 46, 47, 48, 49], some studies of NB convolutional and turbo codes were presented. However, no consistent and comprehensive study has been carried out so far for the design of good NB-TCs.

1.3 Turbo codes in a nutshell

Turbo codes were introduced in 1993 [3] as the parallel concatenation of several (usually two) Recursive Systematic Convolutional (RSC) constituent codes. These codes were shown to approach channel capacity, provided that they are decoded iteratively with an information exchange process between the constituent decoders.

1.3.1 Recursive Systematic Convolutional codes

Convolutional codes were first introduced in 1955 [25] as a linear finite-state machine consisting of a shift register and linear algebraic functions. They are characterized by the number of memory elements ν, which imposes the number of possible states of the encoder as 2^{ν}. The output message of the convolutional code is formed from the redundant information generated by the structure of the code. A convolutional code is systematic when its input message is transmitted alongside with the redundant bits. In [3], the authors have revisited the structure of convolutional codes by defining the Recursive Systematic Convolutional (RSC) codes, consisting of a feedback shift register. The use of the convolutional codes in a recursive form is shown to be essential in the creation of turbo codes [3, 7]. The interest of using RSC codes in the design of turbo codes is explained in the section 1.3.2. Fig. 1.10 shows the structure of a convolutional code with 3 memory elements, which corresponds to an 8-state convolutional code. Data sequence d is encoded with the proposed RSC structure. The output of this code is formed by the systematic part $s = d$ and the calculated parity bits r.

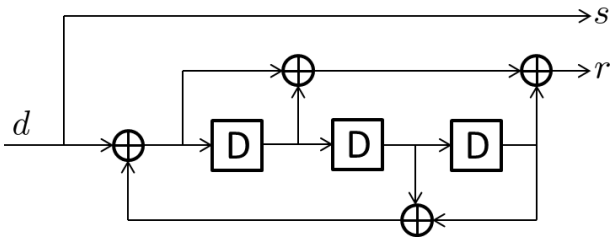

Figure 1.10: Example of a RSC structure. Here, the generator polynomials are $(13,15)_8$ used in the LTE standard.

1.3.2 Parallel concatenation of RSC codes

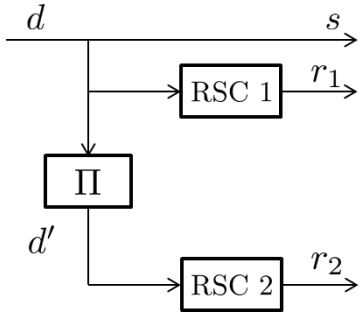

Figure 1.11: Turbo code structure defined as the parallel concatenation of two RSC codes, with a permutation function Π.

Fig. 1.11 shows the basic structure of a turbo encoder, which consists of the parallel concatenation of two RSC codes. The authors in [7] have demonstrated the interest of using recursive codes in the design of TCs. Using a uniform interleaver, non-recursive convolutional encoders are shown to have $\omega_{min} = 1$, where ω_{min} denotes the minimum possible information weight in the ensemble of error events in the considered convolutional code. Information sequences with $\omega_{min} = 1$ are shown to generate error sequences with an identified weight of length $2(\nu + 1)$. Accordingly, it is demonstrated that any information sequence with weight ω can result in the concatenation of ω error events of weight one. The authors have shown that, for the case of non-recursive convolutional constituent codes, the bit error rate probability is independent from the information frame size and then, no interleaving gain can be observed in this case. On the contrary, it is demonstrated that $\omega_{min} \geq 1$ in the case of recursive encoders. In this case, it is shown that the bit error probability of the TC decreases with the increase of the information frame size, and therefore with the interleaver size. This property leads to the conclusion that TCs profit from the interleaver gain when recursive encoders are considered.

The RSC codes are defined by their generator polynomials, chosen in order to ensure that their free Hamming distance is high enough. First, the information frame, denoted by d, is encoded by the first RSC encoder "RSC 1". Then, d is interleaved with the permutation function Π to form the interleaved version of the information frame d' which is encoded using the second RSC encoder "RSC 2". At its output, the turbo encoder provides the systematic bits d with the parity bits generated by the two encoders, r_1 and r_2. A puncturing block can be added to this structure, in order to achieve different code rates, this block is also discussed in Section 1.3.3. The interleaving function π is defined following several constraints that are discussed in Section 1.3.4.

1.3.3 Puncturing turbo codes

The TC structure of Fig. 1.11 essentially produces codewords with coding rate $R = \frac{K}{N} = 1/3$. Higher rates can lead to higher throughput communications, and lower rate will achieve better performance. Depending on the application type, a particular coding rate is targeted. To achieve higher rates, puncturing is applied [11]. Puncturing is applied by disregarding some parity and/or systematic bits at the output of the TC. This elimination is studied for each defined coding rate mutually with the defined interleaving function in order to achieve the highest possible Hamming distance at the output of the TC. When low coding rates are targeted, RSC codes with different parity generator polynomials can be defined in order to generate additional parity bits from each code and then, achieve lower rates.

In some systems, rate compatible encoders should be deployed in order to target different coding rates with acceptable performance. The LTE (Long Term Evolution) standard exploits the rate matching mechanism [29, 50], offering the possibility of achieving incrementally different coding rates. A Circular Buffer Rate Matching (CBRM) algorithm is defined in the LTE standard, which repeats or punctures the bits of the mother codeword in order to generate the requested number of bits according to the desired coding rate.

1.3.4 Turbo code interleaver

The turbo code interleaver is a permutation function that rearranges the information frame d at the input of the constituent encoder "RSC 2". The first role of the interleaver is to efficiently fight against the occurrence of error bursts for at least one of the constituent decoders, at the receiver side. In addition, the design of the interleaver has also a direct impact on the minimum Hamming distance of the turbo code [8]. In order to achieve a high minimum distance, several constraints have to be taken into account in the interleaver design [9, 10].

The first constraint is the spread value, which defines the sum of the position distances in the natural order and in the interleaved order for a couple of symbols:

$$S(i,j) = f(i,j) + f(\Pi(i),\Pi(j)) \tag{1.8}$$

where i and j are the couple positions in the natural order and $f(i,j)$ and $f(\Pi(i),\Pi(j))$ define the distance between the positions i and j in the natural and the interleaved order respectively. The minimum value S_{min} of $S(i,j)$ defines the minimum spreading value of the code. This value shows how neighboring symbols are interleaved to spaced positions due to the interleaving scheme. Besides introducing correlation in the decoding process, permutations with a low minimum spread may lead to low-weight codewords that can limit the minimum Hamming distance of the turbo code. The maximization of S_{min} is of high importance in the design of an interleaver [51]. Fig. 1.12 shows an example of data sequence d interleaved in d' using the interleaving function Π. In this example, we show the connections providing the minimum spreading value. Here $f(0,1) = 1$, $f(\Pi(0),\Pi(1)) = 3$, resulting in $S_{min} = 4$.

The second constraint is related to the correlation cycles in the decoding process. Due to the exchange of extrinsic information between the component decoders, correlation effects can be observed during the decoding process that can lead to the propagation of errors and to a performance penalty [30, 52]. This effect is especially problematic for

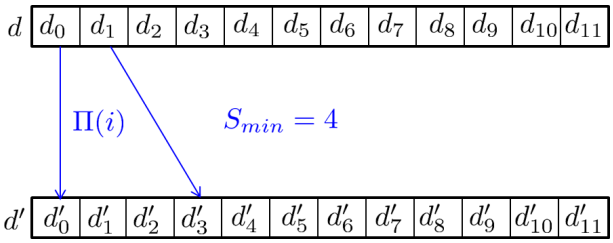

Figure 1.12: An example of data sequence d with frame size $K = 12$. The interleaving function is shown with the positions creating the minimum spreading value S_{min}.

short data frames. Therefore, decreasing correlation in the decoding process, by imposing interleaving constraints, is of high importance [53]. A correlation graph is defined for TCs in [54, 10], where the closed cycles are known as the correlation cycles. The length of the shortest correlation cycle in the correlation graph, known as the girth value g, is shown to have one of the most influential effect on achieved performance. In the example depicted in Fig. 1.13, the extrinsic information generated at position $\Pi(i_1)$ in the natural order, for the information frame d, is driven to position i_1 at the interleaved order, for the interleaved information frame d'. When the recursive forward metric is computed at the interleaved order, position i_3 is directly affected by the information at i_1. At the next iteration, the generated extrinsic information at i_3 is transferred to position $\Pi(i_3)$ at the natural order, and so on. Obviously, after four information exchanges, data at $\Pi(i_1)$ is re-affected by its own generated extrinsic information at the first iteration. Such a cycle is a correlation cycle with length 4.

To limit the detrimental effect of correlation in the decoding process, the length of the correlation cycles should be as high as possible. In conclusion, the value of the girth should be maximized.

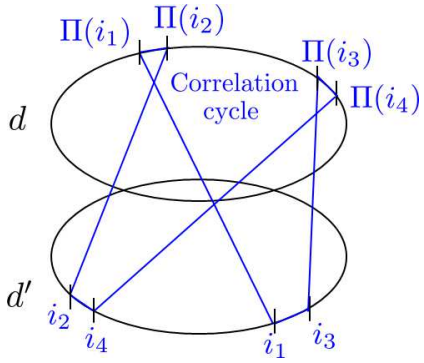

Figure 1.13: Correlation graph example, where the minimum correlation cycle is represented, here $g = 4$.

Most of the interleaver families introduced in the context of TCs were designed according, to at least, one of these constraints. They fall in two main categories: random-based and algebraic interleavers.

1.3.4.1 Random interleaver

Random interleavers were introduced in [55]. The interleaved addresses are chosen ran-

domly with no repetition, where the sequence is written into the interleaver memory block and read out in a specific random order. This interleaver doesn't respect any of the interleaving design criteria described previously. It is shown in [51] that the minimum distance of TCs using random interleavers is governed by sequences with input weight 2. In [56], uniform interleaving is introduced, which is based on a random process that generates all possible permutations, i.e. a different interleaver for each information frame. Uniform interleaving is a useful tool to analyze the average performance of the designed TCs regardless of the interleaving block.

1.3.4.2 Partially Random interleaver

It was shown in [51] that, when using a random interleaver, the low-weight codewords of the turbo code are caused by information sequences with input weight 2. Actually, since no minimum spread is guaranteed with this type of interleaver, the occurrence of error patterns with weight equal to their free distance cannot be avoided for both codes simultaneously and these patterns are caused by weight-2 input sequences. Therefore, spread constraints can be applied to the design of random interleavers in order to reduce the number of such sequences. Partially random interleavers were introduced in this context [57]. They are based on the generation of a random interleaver as a first step. Then, the second step consists in identifying information bit patterns where the smallest distance is observed. In the identified patterns, a bit swapping mechanism is applied in order to break the error-prone sequences. When swapping is verified all along the frame, a new interleaving function is defined, which is shown to have better performance in the error floor region.

Another partially random interleaver was proposed in [51] by Divsalar et al. and called S-random: the data sequence is written into the interleaver memory block and read out in a specific random order like in the random interleaving case. However, in the S-random interleaver, the interleaving addresses are chosen respect the spread property of the code. When respecting high spread values, low-weight codewords are avoided by the design of the interleaver, since neighboring bits in the natural order will enjoy a higher separation distance in the interleaved order. The maximum achievable spreading value is $S_{max} = \sqrt{2K}$, where K is the frame size [58]. The complexity of achieving S_{max} with a random-based interleavers increases with K. It is demonstrated in [51] that choosing a value for the spread that respects $S < \sqrt{\frac{K}{2}}$ can ensure a reasonable simulation time to produce the S-Random interleaver.

1.3.4.3 Algebraic interleaver

Random based interleavers are not well suited to hardware implementations, since a random process is in charge of their generation, and their structure does not offer the ability to design parallel decoders. To this end, algebraic interleavers were defined. These interleavers can be simply implemented based on mathematical equations.

For Regular Interleavers (RI), addresses in the natural order were first defined by:

$$\Pi(i) = (P.i) \bmod K \tag{1.9}$$

where P stands for the interleaver period, $i = 0...K-1$ represents the data symbol address in the interleaved order and $\Pi(i)$ its address in the natural order. When P is chosen carefully respecting certain conditions, the theoretical value of S_{max} can be reached [59, 8]. Turbo codes with RI cannot efficiently achieve very low error rates, since rectangular error patterns are observed in its structure [60]. Rectangular error patterns are composed of multiple Return To Zero (RTZ) sequences that cannot be avoided by the interleaving structure, since the latter is based on a regular structure. To this end, in practice, RI are only used as a basis to define better algebraic permutations.

Different algebraic interleavers have been proposed in the literature: Dithered Relative Prime (DRP) interleavers [61], Quadratic Permutation Polynomial (QPP) interleavers [62], and Almost Regular Permutation (ARP) in [60].

The QPP interleaver was adopted in the the 4G wireless communication systems in 2007 [29], whereas ARP interleavers were adopted in DVB-RCS [28], in WiMAX standard in 2006 [63], and finally in DVB-RCS2 in 2012 [64].

The authors in [12], have shown that the interleaving properties of the DRP and QPP interleavers can be found in an ARP interleaver. It was shown that ARP interleavers can guarantee a minimum Hamming distance at least as high as the distance provided by the QPP and DRP interleavers. For this reason, we focus on the ARP family in our study.

The ARP interleaver is based on a RI with period P, targeting the maximization of the free distance of the TC by achieving high spread values. As shown above, TC with RI interleavers suffer from high error floors due to the introduced rectangular error patters. To this end, additional disorder is included in the ARP interleaver in order to avoid such error patterns. This disorder is introduced by a vector S of Q shift values, where Q stands for the disorder degree. The values of S are defined to respect the requested value of spread S_{min}. The ARP interleaver respects the following equation:

$$\Pi(i) = (P.i + S(i \bmod Q)) \bmod K \tag{1.10}$$

where $i = 0...K-1$ stands for the interleaved address of the data symbol and $\Pi(i)$ its corresponding address in the natural order. The elements of vector S are defined by:

$$S(i \bmod Q) = A(i \bmod Q)P + B(i \bmod Q) \tag{1.11}$$

where A and B are vectors of length Q. $A(i \bmod Q)$ and $B(i \bmod Q)$ are multiples of Q.

In [60], an algorithm is presented in order to find ARP parameters that provide the best spread properties. Firstly, Q is chosen as a divider of the frame size K, depending on the design constraints. Then, the sets of the possible values of P and $\{S(0)...S(Q-1)\}$ are defined so as to respect the requested value of S_{min}. Finally, the maximization of the minimum Hamming distance is taken as the constraint to choose the best set of parameters among the sets respecting the value of S_{min}.

In [10] a new method is proposed for the definition of ARP parameters in order to yield high girth values. To this end, after a random selection of P, the values of the shift vector S are chosen one by one. When a new value of shift $S(i)$ is set, the correlation graph is evaluated. If $S(i)$ results in low correlation cycles, $S(i)$ is reset to a new value. When the Q values are defined, the proposed interleaver is shown to largely improve performance.

A joint optimization between the puncturing and interleaving parameters is studied in [13], where a new method is proposed for the design of ARP interleavers used in punctured TCs, which offers better performance compared to separate designs. The

proposed method begins by defining the puncturing pattern. The best puncturing masks are identified according to their resulting Hamming distance calculated at the component codes. Then, if data puncturing is used, puncturing constraints are identified to be used in the design of the interleaver. These puncturing constraints are based on the sorting of the unpunctured data positions depending on their evaluated error-prone level, and then interleaving connections are proposed in order to connect the least error-prone position to the most error-prone position and so on. After defining the puncturing structure, a group of possible interleavers is defined. The algorithm begins by selecting a value of P in the admissible set of possible values that ensure achieving the targeted spreading value S_{min}. Then, the Q values of the shift vector S are defined one by one. When choosing a value $S(i)$, one should verify if this value respects the puncturing constraints, and if it guarantees the requested values of S_{min} and minimum girth g. If the chosen value $S(i)$ does not respect these two conditions, $S(i)$ is set to another value. Otherwise, $S(i)$ is kept, and the next shift parameter is defined. After defining the set of possible interleavers respecting the minimum values of spread and girth, and respecting the puncturing constraints, the best candidate is chosen as the one having the best distance spectrum. By this method, the authors in [13] have shown ameliorated performance in the error floor regions of TCs, with different coding rates.

1.3.5 Distance spectrum of turbo codes

In the design of TCs, two aspects of the error correction performance are of interest, depending on the application requirements: the waterfall region which defines the convergence behavior of the code, and the error floor region which defines the behavior of the code at high SNRs. Some applications require very low error floors, so that packet re-transmission can be avoided, which enhances the total throughput of the system. The performance in the error floor region of TCs is directly related to their distance spectrum. It is shown in [65] that at high SNR values, the performance of TCs approaches the theoretical limit given by the union bounds defined in [57, 66]. The probability of bit errors (P_b) is upper bounded theoretically by the union bound U_b defined as:

$$P_b \leq U_b = \sum_{d_i=d_{min}}^{d_{max}} \frac{W_{d_i}}{K} Q\left(\sqrt{\frac{2d_i R E_b}{N_0}}\right) \tag{1.12}$$

where d_i represents the different values for the Hamming distances of the code that vary from the minimum distance d_{min} to the highest found distance d_{max}. W_{d_i} stands for the total weight of all codewords at distance d_i. The information frame size and the code rate are represented by K and R respectively. $\frac{E_b}{N_0}$ is the ratio of the bit energy to the noise power spectral density. Q defines the tail distribution function of the standard normal distribution, it can be derived from the error function by $Q(x) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right)$, where $\mathrm{erfc}(x) = \frac{2}{\sqrt{\Pi}} \int_x^\infty e^{-t^2} dt$.

Moreover, the probability of frame error (P_f) is upper bounded by U_f:

$$P_f \leq U_f = \sum_{d_i=d_{min}}^{d_{max}} n(d_i) Q\left(\sqrt{\frac{2d_i R E_b}{N_0}}\right) \tag{1.13}$$

where $n(d_i)$ is the multiplicity of distance d_i, i.e. the number of codewords at distance d_i.

Calculating union bounds can help identify the behavior of the designed code in the error floor region, thus avoiding time and hardware consumption for reliable Monte Carlo performance simulations. To calculate union bounds, the Hamming distance spectrum of the TC should be determined. In the context of binary TCs, several methods were proposed in the literature to estimate distance spectra [67, 68, 69, 70, 71]. All the proposed methods consist in introducing error patterns in the all-zero sequence, and then calculating the Hamming weight of the resulting codeword.

Fig. 1.14 shows a comparison example of BLock Error Rate (BLER) performance of a binary TC, with coding rate $R = 2/3$, an information frame size $K = 208$ bits, and using an ARP interleaver. The scaled Max-Log-MAP algorithm [72, 73] was used for decoding. The distance spectrum for the proposed code is shown in Table 1.1. This example shows the importance of distance spectrum analysis for TCs, since Fig. 1.14 shows that properly designed TCs perform at high SNR close to the calculated union bounds.

d_0	d_1	d_2	$n(d_0)$	$n(d_1)$	$n(d_2)$
8	9	10	26	130	624

Table 1.1: Estimated TC distance spectrum and corresponding multiplicities $n(d_i)$ for the proposed TC, using the 8-state tail-biting RSC $(1, 15/13)_8$ component code, for $R = 2/3$ and $K = 208$.

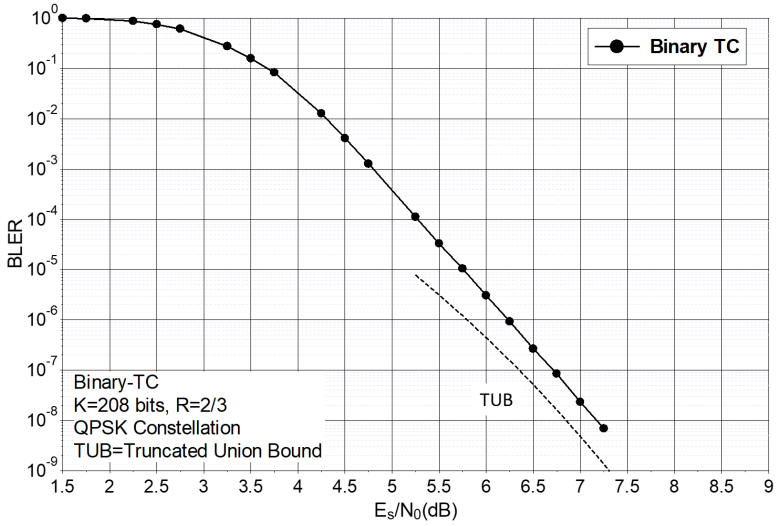

Figure 1.14: Block error rate performance of a binary TC, with $R = 2/3$, $K = 208$ bits, QPSK constellation, ARP interleaver and AWGN channel. Comparison with the truncated union bound (TUB) obtained with the three first distance terms obtained in Table 1.1.

1.4 Conclusion

The necessary background on non-binary error correcting codes and turbo codes is presented in this chapter. The first part introduces the channel coding theory with an

overview of the existing error correction codes as well as the importance of defining new ones.

The second part of this chapter is dedicated to the comparison between non-binary coding schemes and binary coding. The advantages of using non-binary codes are presented, where different motivations are explained mainly based on the advantages of using CM over the BICM.

A review of binary and non-binary LDPC codes is developed. Moreover, a comparison between these codes is conducted in order to assess the importance of studying non-binary coding schemes. Finally, a brief presentation of the structure of TCs is presented, with the description of its different constituent blocks.

This chapter shows interesting motivations for the definition of non-binary error correction codes, namely NB-TCs. Surprisingly, no general structure for NB-TCs is proposed in the literature. The aim of this thesis is to define such coding scheme, hopefully satisfying at least parts of the requirements of future standards listed in this chapter.

Chapter 2

Design of convolutional codes over GF(q)

In Chapter 1, we showed that Non-Binary (NB) codes have a real potential to outperform their binary counterparts, especially when they are associated with high-order modulations and for the transmission of small data blocks.

Non-binary Low Density Parity-Check (NB-LDPC) codes have been widely studied in the literature. They show remarkable performance and extensive studies have been carried out to reduce their decoding complexity in order to pave the way for hardware implementations with up-to-date semiconductor technology.

In contrast, research related to NB convolutional and turbo codes over Galois fields is very limited and a general study of the design of convolutional codes to be used as component codes for NB Turbo Codes (NB-TC) over Galois fields seems to be missing in the literature.

In this chapter, the first step towards defining NB-TCs over Galois fields, that is, the definition of NB Convolutional Codes (NB-CC), is described. The chapter is organized as follows: Section 2.1 summarizes some basics about Galois fields. Then, a state of the art of NB-CCs and NB-TCs is presented in Section 2.2. Section 2.3 and Section 2.4 describe the main contributions of the chapter, that is, the structure of the studied NB-CC, along with the underlying logic, the method applied to select the generator polynomials of the codes and the resulting codes. Section 2.4 also proposes a low-complexity method to compute the cumulated Euclidean distance spectrum of a NB-CC, discusses the impact of the constellation mapping on the code selection and provides some simulation results. Additionally, Section 2.5 presents a technique based on constellation downsizing that can improve the correction performance of the coded modulation while reducing the demodulation complexity. Finally, Section 2.6 deals with the adaptation of the NB-CCs when they are associated with binary modulations and Section 2.7 concludes the chapter.

2.1 Preliminaries on the calculation in Galois fields

Definition: A Galois Field GF(q) is a finite field that contains q elements, with $q = p^m$, where p is a prime number and m a positive integer. In practical coding applications, p is commonly taken equal to 2. A Galois field GF(2^m) can be constructed using a polynomial $\phi_m(x)$, called ***primitive polynomial***, provided that:

1. the coefficients of $\phi_m(x)$ belong to GF(2), i.e. are binary,

2. the degree of $\phi_m(x)$ is m, which defines the order of the finite field,

3. $\phi_m(x)$ is irreducible over GF(2).

The field GF(2^m) can then be defined as the set of residues modulo $\phi_m(x)$, i.e the set of all polynomials of degree $n < m$.

If α denotes a primitive element of the field, i.e α is a non-zero root of the primitive polynomial $\phi_m(x)$ ($\phi_m(\alpha) = 0$), every non-zero element of GF(2^m) can be written as α^k, $k = 0 \ldots 2^m - 1$ and GF(2^m)=$\{0, 1, \alpha, \alpha^2, ..., \alpha^{2^m-2}\}$, with $\alpha^{2^m-1} = 1$ since $\phi_m(x)$ divides $x^{2^m-1} + 1$ and $\phi_m(\alpha) = 0$. In other words, the powers of a primitive element generate all the non-zero elements of GF(2^m).

Taking GF(4) as an example, the only primitive polynomial of degree 2 is $\phi_2(x) = x^2 + x + 1$. Since the four elements of GF(4) can be written as polynomials with degree $n < 2$, they can be represented in the form $a\alpha + b$, where α is the considered primitive element, while a and b belong to GF(2). Therefore, since $\phi_2(\alpha) = 0$, $\alpha^2 = \alpha + 1$ and GF(4) = $\{0, 1, \alpha, \alpha^2\}$ can also be expressed as GF(4) = $\{0, 1, \alpha, 1 + \alpha\}$.

Table 2.1 shows examples of primitive polynomials that can be used for the definition of different GF(q).

m	ϕ_m
2	$x^2 + x + 1$
3	$x^3 + x^2 + 1$
4	$x^4 + x^3 + 1$
5	$x^5 + x^3 + 1$
6	$x^6 + x^5 + x^3 + x^2 + 1$
7	$x^7 + x^3 + 1$
8	$x^8 + x^4 + x^3 + x^2 + 1$
9	$x^9 + x^4 + 1$
10	$x^{10} + x^3 + 1$

Table 2.1: Examples of primitive polynomials for different GF(2^m).

Considering the example of GF(2^3), $\phi_3 = x^3 + x^2 + 1$. The elements in GF(8) are defined by $\{0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$. Besides, they can also be represented by polynomials of degree 2 : $a\alpha^2 + b\alpha + c$, where a, b and $c \in$ GF(2). Table 2.2 shows the representation of the elements in GF(8) according to the chosen primitive polynomial. The leftmost bit corresponds to the most significant bit in the adopted binary representation, while the rightmost bit corresponds to the least significant bit.

Element in GF(8)	Polynomial representation	Binary value
0	0	000
1	1	001
α	α	010
α^2	α^2	100
α^3	α^2+1	101
α^4	α^2+α+1	111
α^5	α+1	011
α^6	α^2+α	110

Table 2.2: Representation of the elements in GF(8). The construction of this field is done using ϕ_3 from Table 2.1.

Next, we define the main arithmetic operations in GF, specifically the addition and multiplication.

Addition in GF: Addition in GF is straightforward; it is a bitwise addition of the coefficients in the polynomial representation of the numbers. Example in GF(64) :

$$\begin{aligned}(41)_{10} + (12)_{10} &= (101001)_2 + (001100)_2 \\ &= (100101)_2 = (37)_{10}\end{aligned} \tag{2.1}$$

Multiplication in GF: The result of the multiplication of two values in the Galois fields is obtained by the following steps :

- multiplication of the polynomial representation of the two numbers,
- division of the result by the primitive polynomial,
- the remainder of this division is the result of the multiplication.

Example: in GF(64), let the primitive polynomial = $109_{10} = (1101101)_2$ and let the symbol " $\bullet$ " denote the multiplication,

$$(41)_{10} \bullet (12)_{10} = (101001)_2 \bullet (001100)_2 \tag{2.2}$$

we begin by the multiplication of the polynomial representation of the two values :

$$(x^5 + x^3 + 1) \bullet (x^3 + x^2) = x^8 + x^7 + x^6 + x^5 + x^3 + x^2. \tag{2.3}$$

Then, we divide this result by the primitive polynomial, and take the remainder of this division as the result of the multiplication in GF(64):

$$\begin{aligned}&(41)_{10} \bullet (12)_{10} \\ &= (x^8 + x^7 + x^6 + x^5 + x^3 + x^2) modulo \\ &(x^6 + x^5 + x^3 + x^2 + 1) \\ &= x^5 + x^4 + x^2 + 1 = (110101)_2 = (53)_{10}.\end{aligned} \tag{2.4}$$

2.2 State of the art on non-binary convolutional and turbo codes

In the past, very few studies were conducted in the domain of non-binary convolutional codes (NB-CCs) and turbo codes (NB-TCs). In this section, we provide a detailed study about the state of the art in the domain of $m-binary$ codes, NB-CCs and NB-TCs.

2.2.1 $m-binary$ convolutional and turbo codes: an intermediate design between binary and non-binary codes

In [31, 74], $m-binary$ turbo codes were designed based on the NB-CCs defined in [75]. In this proposal, m bits enter the convolutional code structure of ν memory elements, with $m \leq \nu$. The data bits are connected to the possible entries of the CC via a connection grid, which is defined jointly with the generator polynomial of the CC in order to design a code achieving a Hamming distance as high as possible.

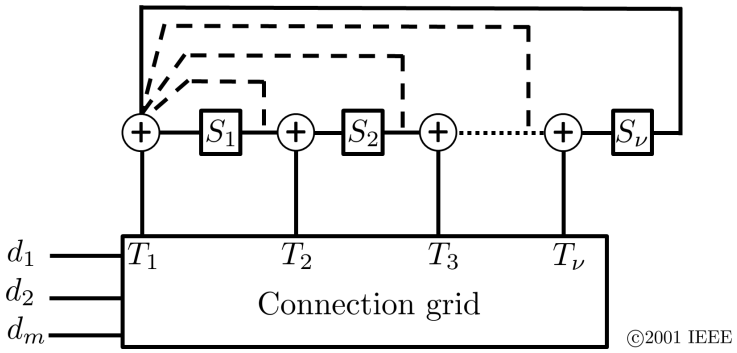

Figure 2.1: The $m-binary$ CC structure with memory ν as proposed in [31, 74].

The authors of [31] have stated several advantages that could be obtained with $m-binary$ TCs based on the component encoders in Fig. 2.1:

1. Better convergence: in [76, 31], it was shown that the correlation is reduced when $m-binary$ codes are defined with respect to binary codes, and this reduction is found to be interesting when $m=2$. Double-binary codes [31, 74], defined with 2 input bits ($m=2$), have shown better convergence when compared to binary codes, due to lower correlation problems in the decoding process.

2. Larger minimum distance: a new degree of freedom is introduced in the $m-binary$ TCs, intra-symbol interleaving. When well designed, the total interleaving function, which is composed from an inter-symbol and an intra-symbol permutations, can help the code achieve high Hamming distances. Therefore, better performance can be expected.

3. Less sensitivity to puncturing: when high coding rates have to be achieved, m–$binary$ TCs offer the possibility of lower parity puncturing, which reduces the performance degradation when increasing the coding rate.

4. Reduced latency: obviously, TCs based on the convolutional codes of Fig. 2.1 perform the decoding of m bits at each trellis stage, and thus, lower latency can be achieved.

5. Robustness of the decoder: it is shown in [31] that the gap in performance between the reduced-complexity decoder and the full MAP decoder is reduced when *m–binary* codes are introduced. The authors in [31] suppose that this property is due to the reduction of the number of trellis stages for the same information frame size K, and then the decoder can perform closer to the Maximum Likelihood (ML) decoder.

Consequently, the design proposed in [31, 75] offers many coding advantages in comparison with binary TCs. Also, the performance of double-binary TCs, for different coding rates, was provided in [31] and they showed interesting gains when compared with binary TCs. These promising advantages and performance justified the interest of studying non-binary TCs. The proposed codes in [75, 31] are not defined over rings, groups or fields, and data processing in their structures remains at the bit level. NB-TCs designed over algebraic structures, where information data are processed in the symbol domain, can offer more degrees of freedom and thus may yield better performance especially when mapped to their corresponding constellation as stated in Section 1.2.3.

2.2.2 Convolutional codes over rings

NB-CCs over rings were firstly introduced in [77], where they are shown to offer good performance with phase modulations. Many studies have further investigated such structures, e. g. [78, 79, 80, 81].

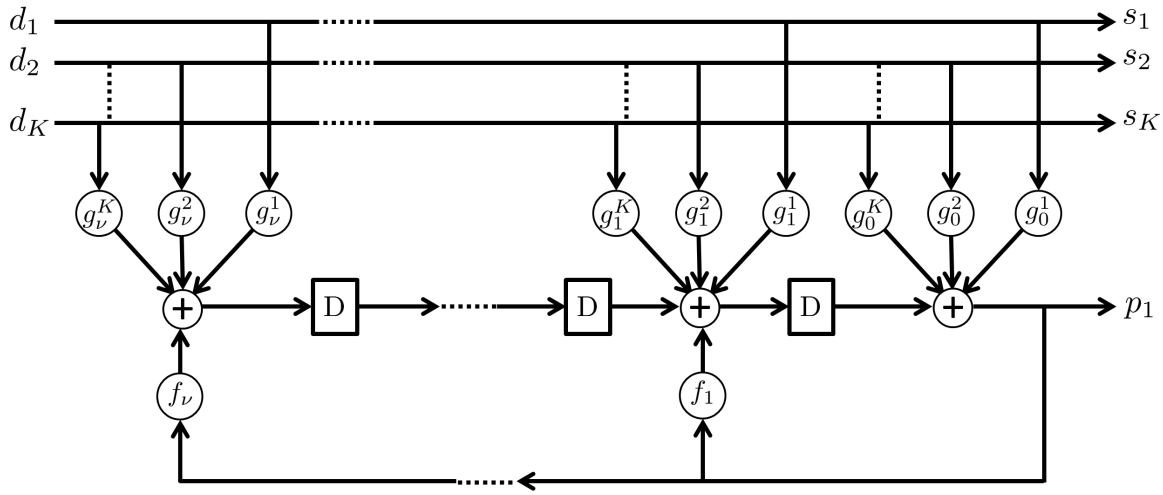

Figure 2.2: The general structure of NB-CCs over rings as proposed in [78].

The structure proposed in [78] is depicted in Fig. 2.2, where a NB-CC over rings with ν memory elements is shown, with the generation of one parity for every K symbols. Its coding rate is $R = K/(K+1)$. The feedforward and feedback generator polynomials are represented by the coefficients g_D^i and f_D respectively. In polynomial form: $f(D) = f_1 D + ... + f_\nu D^\nu$ and $g^i(D) = g_0^i + g_1^i D + ... + g_\nu^i D^\nu$, with $1 \leq i \leq K$.

The generator matrix of such a code, $G(D)$, can be written as :

$$G(D) = \begin{bmatrix} & \vdots & g^1(D)/f(D) \\ I_K & \vdots & \vdots \\ & \vdots & g^K(D)/f(D) \end{bmatrix} \tag{2.5}$$

where I_K stands for the identity matrix that is responsible for the systematic property of the code. The main design goal is the identification of the coefficients f_D and g_D^i in the used mathematical ring, which results in the maximization of the minimum value of the cumulated Euclidean distance. The resulting codes have shown interesting performance when designed with Phase-Shift-Keying (PSK) modulations. This is mainly due to the fact that symbols defined in a ring of integers modulo-m are invariant to $2k\pi/m$ phase rotations. This property avoids error propagation when an m-PSK constellation is used.

The structure of the generator matrix for NB-CCs over rings is well investigated in [79, 80], where the authors have identified several properties that can help define good codes. Conditions on the invertibility, the systematic property, and non-catastrophic codes are presented. The proposed RSC codes over rings are well suited for PSK constellations. However, their structure does not target QAM modulations. Moreover, no studies were conducted in the literature in order to design NB-TCs based on the component codes of Fig. 2.2.

Besides their attractive behavior when designed with m-PSK constellations, Konishi has proposed a design of NB-CCs over rings well suited for QAM modulations [81]. This new structure is composed of one or two memory elements as shown in Fig. 2.3. This structure, which is non-systematic, is shown to be well suited to the context of QAM constellations when the coefficients a_i^j are well chosen. Multiplications and additions are applied over rings modulo m, corresponding to the order of the used m-QAM modulation.

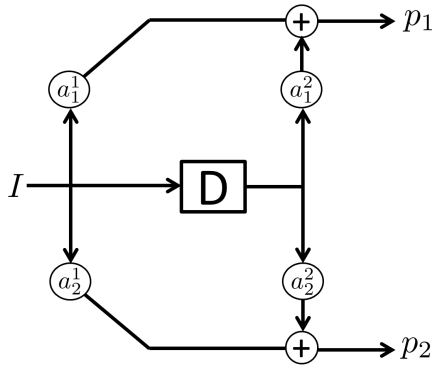

Figure 2.3: Non systematic NB-CCs over rings with one memory element, as proposed in [81].

In [81], the author has designed NB-CCs over ring $\mathbb{Z}_{64}$ well suited for 64-QAM modulations. His approach is based on a extended signal set that takes advantage of the matched mapping property for the computation of the Euclidean distances.

The main drawback of rings such as $\mathbb{Z}_{64}$ is the presence of non-zero divisors in the ring, which limits the minimum distance achieved by the codes. Consequently, the author of [81] proposes to use irregular multiplication to circumvent this problem. Good codes with high Euclidean minimum distances are found using this technique. However, the proposed structure is not suitable for the design of NB-TCs since it is a non-systematic and non-recursive structure.

2.2.3 Turbo codes based on time-variant non-binary convolutional codes

In [44, 45], two novel structures of NB-TCs are proposed based on the design of time-variant NB-CCs. The proposal consists in deriving a protograph sub-ensemble of a regular ($d_v = 2$, $d_c = 3$) NB-LDPC code designed based on [37]. A protograph is defined as a Tanner graph [36], having a small number of connections, consisting of the set of variable nodes (VNs), the check nodes (CNs) and the set of edges ϵ. When expanded, with a copy-and-permute procedure, a protograph can result in the Tanner graph of the total NB-LDPC code. It is described by a base matrix B:

$$B = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & 2 \end{bmatrix} \tag{2.6}$$

When expanding B, the resulting NB-LDPC matrix can be written as:

$$H = \begin{bmatrix} \tilde{I} & \tilde{P}^{(1)} & 0 \\ \tilde{\Pi} & 0 & \tilde{P}^{(2)} \end{bmatrix} \tag{2.7}$$

where $\tilde{I}$ is a $K \times K$ diagonal matrix, which elements define polynomial $g^{(1)}$. From the $K \times K$ matrix $\tilde{\Pi}$, the generator polynomial $g^{(2)}$ and the permutation functions are defined, since $\tilde{\Pi}$ is defined as the product between the diagonal matrix having the coefficients equivalent to the elements of the polynomial $g^{(2)}$ with the permutation matrix Π. Finally, $\tilde{P}^{(1)}$ and $\tilde{P}^{(2)}$ define $f^{(1)}$ and $f^{(2)}$. The authors in [44, 45] have shown that $\tilde{P}^{(1)}$ and $\tilde{P}^{(2)}$ can be written as:

$$\tilde{P}^{(i)} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & f_0^{(i)} \\ f_1^{(i)} & 1 & 0 & \dots & 0 & 0 \\ 0 & f_2^{(i)} & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & f_{K-1}^{(i)} & 1 \end{bmatrix} \tag{2.8}$$

$g^{(1)}$, $g^{(2)}$, $f^{(1)}$ and $f^{(2)}$ define the generator polynomials of the two constituent encoders respecting the structure of Fig. 2.4.

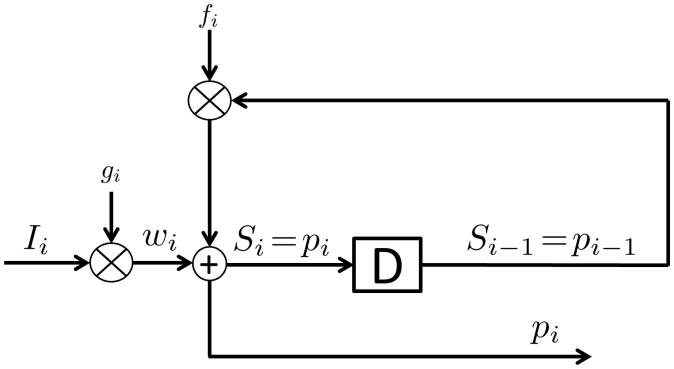

Figure 2.4: The structure of memory-1 time-variant NB-CC as proposed in [44, 45].

Two techniques were proposed for the design of the interleaving function Π, either by respecting turbo code constraints by increasing the interleaver spread, or by respecting

the girth optimization in the NB-LDPC cycle graph defined by $\boldsymbol{H}$. The authors have shown that both methods result in the design of good codes.

The generator polynomials, $g^{(1)}$, $g^{(2)}$, $f^{(1)}$ and $f^{(2)}$, are defined according to the construction method of the parity-check matrix $\boldsymbol{H}$ proposed in [37].

Two possible code classes are considered in [44, 45]. Firstly, a parallel concatenation of the devised CCs with the permutation function is proposed, targeting the structure of a NB-TC. Secondly, the protograph is stretched serially resulting in the concatenation a memory-1 non recursive encoder, an interleaver, and a NB-RSC code.

2.2.3.1 Parallel concatenation

The matrix $\boldsymbol{H}$ resulting from Eq. (2.7) can be seen as the parity-check matrix of a NB-TC, based on the NB-CC of Fig. 2.4 and on the designed interleaving function Π. The resulting NB-TC is depicted in Fig. 2.5

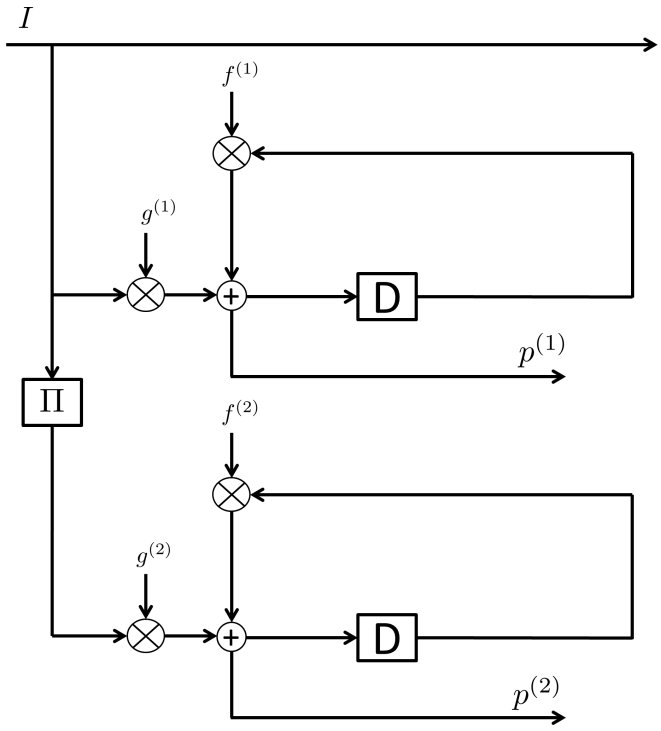

Figure 2.5: The structure of a NB-TC as proposed in [44, 45].

When respecting the NB-CC design of Fig. 2.4, and the constraints of [37], the generator polynomials and the interleaving function of Fig. 2.5 are defined so as to achieve attractive error correcting performance for different coding rates.

2.2.3.2 Serial concatenation

An alternative construction to the one provided in 2.2.3.1 was also proposed [44, 45], based on a serial concatenation rather than a parallel one.

The protograph created from the NB-LDPC code is stretched to form a serial structure. The new scheme can be seen as a serial concatenation between a non-systematic code, an interleaver and a RSC code.

In [44, 45], the serial concatenation results in coding rate $R = 1/2$, since the resulting parities $p^{(1)}$ generated by the first encoder are encoded by the second, and the output codeword is $c = [I|p^{(2)}]$. When rate $R = 1/3$ is targeted, the authors propose to set the output codeword as $c = [I|p^{(1)}|p^{(2)}]$. It is demonstrated that the resulting structure is equivalent to a non-binary irregular repeat accumulate (IRA) code with matrix:

$$H = \left[\begin{array}{cc|cc} \tilde{\Pi}^{-1} & \tilde{P}^{(2)} & \tilde{I}^{-1} & \tilde{P}^{(1)} \end{array} \right] \tag{2.9}$$

The codes described in Sections 2.2.3.1 and 2.2.3.2 have shown interesting results: gains of more than 1 dB are found when compared to binary LDPC and turbo codes. Also, the authors of [44, 45] have shown that the proposed codes can be decoded by either the LDPC decoding algorithms, or by the turbo decoding algorithms. Furthermore, these studies offer the first designed NB-TCs over high order fields GF(q) with $q > 16$, while other studies are limited to NB-TCs targeting low order finite fields.

Nevertheless, this proposal does not benefit from the offered degrees of freedom inherent to the structure of NB-TCs, since it is constructed based on the parity-check matrix of NB-LDPC codes. Also, the joint optimization of NB codes with the corresponding constellation is not targeted by this design.

2.2.4 Non-binary convolutional and turbo codes for impulsive noise channels

In [46], the design of NB-CCs over GF(4) is investigated. A structure similar to the one shown in Fig. 2.3 is used, with more than one memory element. The proposed structure is non-systematic and non-recursive, the data entries I_i are in the considered GF(q), and the generator polynomials a_i^1, a_i^2 are defined in order to optimize the waterfall performance, targeting Physical-Layer Network Coding (PNC) over an impulsive conventional Two-Way Relay Channel (TWRC). In this paper, a comparison is conducted with 4-state binary convolutional codes. It is shown that NB-CCs outperform the binary codes by more than 1 dB. No further design clarifications are shown in this paper, as for instance the technique for the identification of the generator polynomials.

In [47], Zhao *et al.* have used the structure from [46] to design NB-TCs over GF(4). However, since the NB-CCs in [46] are non-recursive, they have proposed a modified structure of NB-CCs over GF(4) as shown in Fig. 2.6. This structure is systematic, with a feedback coefficient equal to 1. The feed-forward coefficients are β and β^2, where β is a primitive element in GF(4).

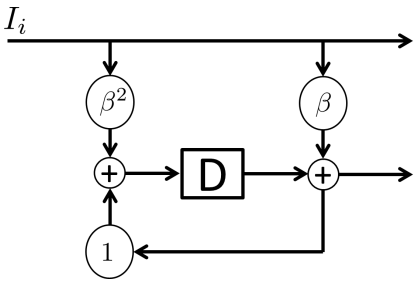

Figure 2.6: The structure of NB-CCs over GF(q), with 1 memory element, as proposed in [47].

In [47], only the analysis of NB-TCs over GF(4), with a specific channel type, namely additive impulsive noise channels, is provided. A more general study of NB-TCs over GF(4) is proposed in [48, 49], using the same encoding template as in Fig. 2.6.

All these papers show promising results for NB-TCs over GF(4) when compared with their binary counterparts. The authors in [48] have shown that these codes offer better performance than binary TCs due to their capability of resisting burst errors. Furthermore, the additional complexity imposed by the non-binary structure can be partly remedied since NB-TCs are shown to converge to the optimal performance in less iterations than binary TCs. In these papers, the study of NB-TCs over different Galois fields, and the optimization of the interleaving function, are not addressed.

The main contributions about NB-CC and NB-TC found in the literature have been gathered in [82].

2.3 Proposed structures of non-binary recursive systematic convolutional codes

Different structures can be proposed in the context of NB-CCs over GF(q), with $q = 2^m$ and m is a positive integer. In our work, the goal is to propose new encoders that do not show high design complexity when targeting a hardware implementation of the decoders. To this aim, convolutional codes with one memory element were considered in this work, thus limiting the number of states in the code trellis, while providing good decoding performance. Inspired from the binary case, the accumulator structure (structure S_1 in Fig. 2.7), was firstly proposed and used as an encoding template. Then, a more general and elaborated structure was proposed, denoted by S_2 (Fig. 2.7). S_2 was the one adopted in our work due to different reasons that will be presented shortly.

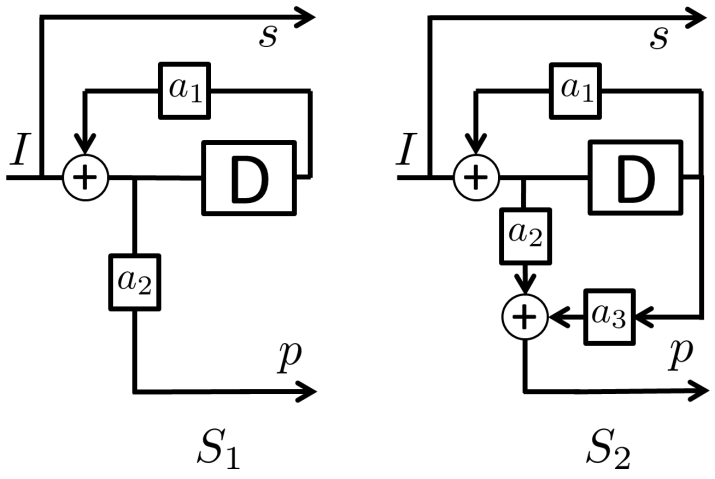

Figure 2.7: Non-binary convolutional code structures with one memory element.

In the proposed models of Fig. 2.7, one parity is generated from each structure, and the proposed codes have coding rate $R = 1/2$.

For a code defined over GF(q), coefficients a_1, a_2, a_3, the input data I, systematic and parity output data, s ($s = I$) and p, are elements of GF(q). Since the encoding structures use only one memory element, the codes considered here have the lowest level

of complexity, for a given value of q, among all the codes over GF(q). Their trellis diagrams have a number of states equal to the order q of the Galois field and are fully connected.

As already shown in Section 1.3.2, the use of RSC encoders is essential to the construction of TCs in order to benefit from the interleaving gain offered by the internal permutation. To this end, structures S_1 and S_2 are recursive and systematic. The recursion polynomial, denoted by $P_r(D)$, is:

$$P_r(D) = 1 + a_1 D \tag{2.10}$$

In order to respect the recursive nature of the code, $P_r(D)$ should not be a constant polynomial. This condition directly leads to the constraint that $a_1 \neq 0$.

The structures in Fig. 2.7 differ in the parity polynomials. For structure S_1, it is equal to:

$$P_{p_1}(D) = a_2 + a_2 a_1 D \tag{2.11}$$

and for structure S_2 to:

$$\begin{aligned} P_{p_2}(D) &= a_2(1 + a_1 D) + a_3 D \\ &= a_2 + (a_1 a_2 + a_3)\, D \end{aligned} \tag{2.12}$$

Just as the feedback polynomial, the parity polynomial should also not be a constant polynomial. To this end, an additional constraint should be met in every structure. For structure S_1, it is sufficient to assume that $a_2 \neq 0$. For S_2, we should have:

$$a_1 a_2 + a_3 \neq 0 \tag{2.13}$$

When $a_1 a_2 + a_3 = 0$, the input data is just multiplied by a fixed coefficient without being encoded.

Structure S_2 is more general than S_1 since the latter can be derived from the former by using the same coefficients a_1 and a_2 and by setting $a_3 = 0$.

Let E_i and E_{i+1} be the encoder states at time i and $i+1$ and s_i and p_i be the systematic and parity symbols labeling the transition between E_i and E_{i+1}. For both structures, the relation between the successive encoder states is:

$$E_{i+1} = s_i + a_1 E_i \tag{2.14}$$

For structure S_1, the parity symbol labeling the transition between E_i and E_{i+1} can be written as:

$$p_{1,i} = a_2 (s_i + a_1 E_i) = a_2 E_{i+1} \tag{2.15}$$

For structure S_2, it is written as:

$$p_{2,i} = a_2 (s_i + a_1 E_i) + a_3 E_i = a_2 E_{i+1} + a_3 E_i \tag{2.16}$$

If we assume that coefficients a_1, a_2 and a_3 are non-zero, $(q-1)^2$ different NB convolutional codes can be obtained with structure S_1 when varying the coefficients, while structure S_2 can provide $(q-1)^3$ different codes. Fig. 2.8 illustrates two trellis examples over GF(4), for S_1 and S_2. For both structures, Eq. (2.14) shows that for a given value of a_1, q transitions stem from each state E_i, ending in q different possible states E_{i+1}. Eq.

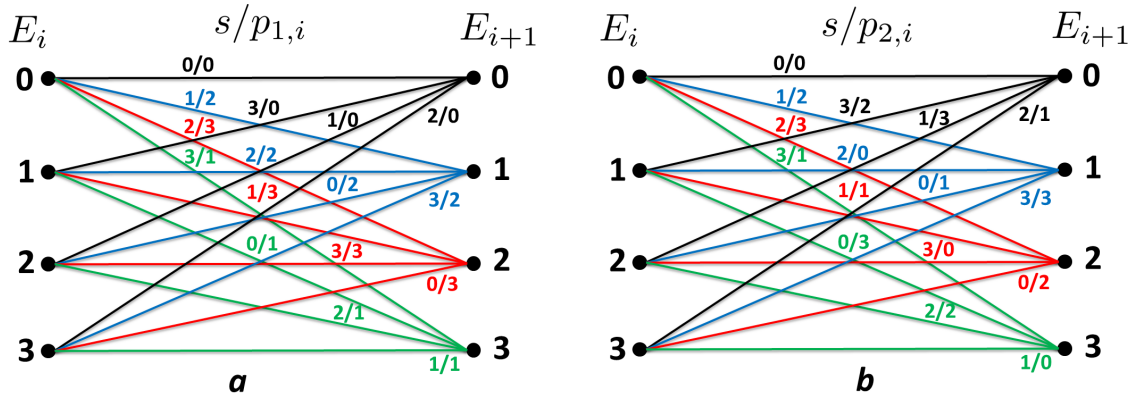

Figure 2.8: Examples of fully connected trellises over GF(4) for: (a) NB-CC structure S_1, (b) NB-CC structure S_2.

(2.14) also shows that these q transitions are labeled with q different values of systematic symbols. Eq. (2.15) or (2.16) provide the corresponding q different values of parity symbols, for each of the parity polynomials, and for the two structures considered in this study. The main difference between structures S_1 and S_2 is that the parity symbol $p_{1,i}$ is the same for the q transitions arriving at state E_{i+1} in S_1 due to Eq. (2.15) (Fig. 2.8-a). On the contrary, Eq. (2.16) shows that a non-zero value of a_3 allows all the transitions arriving at state E_{i+1} in S_2 to be labeled with q different parity values $p_{2,i}$ (Fig. 2.8-b). Consequently, structure S_2 allows the search space for the code to span all the combinations of systematic and parity symbols for each transition between two states in the trellis. To this end, S_2 is seen as a more general structure than S_1, and is adopted in our work as the encoding template.

2.3.1 Structure of non-binary RSC code targeting low rate codes

When low rate codes are targeted with $R < 1/2$, more parities can be generated from S_2. In Fig. 2.9, structure S_2 is proposed with n different parities, resulting in a RSC code with rate $R = 1/(1+n)$.

The parity polynomial $P^m_{p^m}(D)$ corresponding to the parity with index m can be written as, for $1 \leq m \leq n$:

$$\begin{aligned} P^m_{p^m}(D) &= a^m_2(1 + a_1 D) + a^m_3 D \\ &= a^m_2 + \left(a_1 a^m_2 + a^m_3\right) D \end{aligned} \tag{2.17}$$

and parity p^m is provided by the following equation:

$$p^m_i = a^m_2 (s_i + a_1 E_i) + a^m_3 E_i = a^m_2 E_{i+1} + a^m_3 E_i \tag{2.18}$$

In order to have a valid $R = 1/(1+n)$ NB-CC, parity polynomials $P^m_{p^m}(D)$ should not be constant polynomials, $P^m_{p^m}(D) \neq 0$, which imposes that $a^m_2 \neq 0$ or $a_1 a^m_2 + a^m_3 \neq 0$, thus $a^m_2 + a^m_3 \neq 0$. When a^m_2 or a^m_3 is null, a NB-CC with $R = 1/(1+n)$ is defined respecting structure S_1 (when a^m_3 is null), or an equivalent structure to S_1 with one connection providing the parity polynomial (when a^m_2 is null). Since a_1 is independent from m, and a^m_2 and a^m_3 take q possible values each, while the combination $(a^m_2, a^m_3) = (0,0)$ is excluded, then the legitimate number of combinations for (a^m_2, a^m_3) is $q^2 - 1$. In conclusion, each

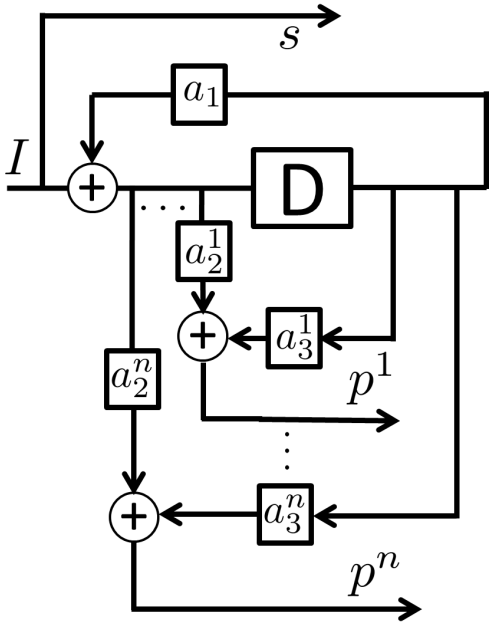

Figure 2.9: Non-binary convolutional code structure S_2 with n different generated parities.

parity polynomial can take q^2-1 possible values. However, we should respect that $P^m_{p^m}(D) \neq P^r_{p^r}(D)$, with $r \neq m$, in order to avoid parity repetitions. Assuming that the number of parity polynomials n verifies the condition $n \leq q^2-1$, the number of possible permutations of n different polynomials is $\frac{(q^2-1)!}{(q^2-1-n)!}$. By taking into account the $q-1$ possible values of a_1, $(q-1)\frac{(q^2-1)!}{(q^2-1-n)!}$ different codes can be obtained with n parity polynomials. Also, the maximum number of distinct parity polynomials that could be provided by this structure is $n_{max}=q^2-1$.

2.4 Selection of the generator polynomials for memory-1 convolutional codes over GF(q), $q > 2$

The task of defining component NB-CCs over different GF(q) is essential for the construction of NB-TCs. In order to find the best NB-CC over each finite field, one should define the main criterion that helps identifying the best corresponding generator polynomials. First, in section 2.4.1, we discuss the criterion for the selection of the NB-CC polynomial coefficients in each finite field GF(q). Then, section 2.4.2 describes a low-complexity method to compute the cumulated Euclidean distance spectrum of a NB-CC code. Section 2.4.3 discusses the impact of the constellation mapping on the code selection. The results of the search for NB-CC generator polynomials are presented in sections 2.4.4 and 2.4.5 for different Galois field orders and coding rates, where the designed codes are also evaluated with Monte-Carlo simulations, in order to validate the proposed design approach.

2.4.1 Design Criterion

As explained by Ungerboeck in [83], when codes are designed to be associated with a high-order modulation, the evaluation of the Euclidean distance spectrum is used as the selection criterion for a code, instead of the Hamming distance spectrum.

On the other hand, the selection of a NB-CC is usually based on the minimization of the average symbol error probability P_s, which is upper bounded by [84]:

$$P_s \leq \sum_{I \in \mathcal{S}} \sum_{\hat{I} \in \mathcal{S}} n(I, \hat{I}) p(I) P(I \rightarrow \hat{I}) \tag{2.19}$$

where $\mathcal{S}$ denotes the set of different possible sequences in the trellis, I and $\hat{I}$ are the information sequences corresponding respectively to the correct path and to an erroneous path in the trellis, $p(I)$ is the probability that the source transmits sequence I, $P(I \rightarrow \hat{I})$ is the Pairwise Error Probability (PEP), that is, the probability that the decoder chooses erroneous sequence $\hat{I}$ instead of the correct transmitted sequence I, and $n(I, \hat{I})$ is the number of erroneous symbols due to such an error event. For the case of an AWGN channel, deriving the Chernoff bound for the PEP [85, 86] gives:

$$P(I \rightarrow \hat{I}) \leq e^{-\frac{E_s}{4N_0} \sum_{n=1}^{N} \left| I_n - \hat{I}_n \right|^2} \tag{2.20}$$

where E_s is the average energy per transmitted symbol for the considered constellation $\mathcal{C}$. Eq. (2.20) shows that the maximization of the Euclidean distance between sequences in the trellis diagram results in the minimization of the average symbol error probability. Therefore, the NB-CC selection can be performed according to the maximum Euclidean distance criterion.

When binary convolutional codes are considered, the minimum Hamming distance and the first terms of the distance spectrum are calculated by assuming that the all-zero sequence has been transmitted, and by calculating the Hamming distances between the all-zero sequence and all the Return To Zero (RTZ) sequences. When a NB-CC code is considered, the same approach to compute the Euclidean distance spectrum can only be applied if the constellation is geometrically uniform and has therefore the uniform error property [84]. Otherwise, the signals generated by the modulator do not all have the same error probability and the all-zero sequence can no longer be taken as a reference. Unfortunately, most of standard high-order digital modulations (q-QAM, q-APSK, ...) do not have the uniform error property. Therefore, to determine the distance spectrum of the corresponding coded modulation, we have to consider all possible pairs of coded sequences, with paths diverging from a given state in the trellis diagram and then converging again to a given state. We call such sequence pairs DC (diverging and converging) sequences or paths. The corresponding squared cumulated Euclidean distance (CSED) is computed as the sum of the Euclidean distances between symbols transmitted along the two DC paths.

For DC paths stretching over L trellis sections, the CSED between two DC sequences

X^1 and X^2 is calculated as follows:

$$\begin{aligned} D^2_{Euc} &= \sum_{l=1}^{L} \left(d^2(X^1_{ls}, X^2_{ls}) + d^2(X^1_{lp}, X^2_{lp}) \right) \\ &= \sum_{l=1}^{L} \Big[(I_{X^1_{ls}} - I_{X^2_{ls}})^2 + (Q_{X^1_{ls}} - Q_{X^2_{ls}})^2 \\ &\quad + (I_{X^1_{lp}} - I_{X^2_{lp}})^2 + (Q_{X^1_{lp}} - Q_{X^2_{lp}})^2 \Big] \end{aligned} \tag{2.21}$$

where X^b_{ls} and X^b_{lp} are the systematic and parity values respectively, at trellis section l in sequence X^b, $b = 1,2$, and I_x and Q_x represent the in-phase and quadrature components of constellation signal x.

The search for good NB-CCs consists in searching for the set of coefficients a_1, a_2 and a_3 in Fig. 2.7 that maximize the lowest D_{Euc} values while minimizing their multiplicities (i.e. the number of sequence pairs with a given distance).

When rates lower than 1/2 are considered following Fig. 2.9, Eq. (2.21) becomes:

$$\begin{aligned} D^2_{Euc} &= \sum_{l=1}^{L} \left(d^2(X^1_{ls}, X^2_{ls}) + \sum_{m=1}^{n} d^2(X^1_{lp^m}, X^2_{lp^m}) \right) \\ &= \sum_{l=1}^{L} \Big[(I_{X^1_{ls}} - I_{X^2_{ls}})^2 + (Q_{X^1_{ls}} - Q_{X^2_{ls}})^2 \\ &\quad + \sum_{m=1}^{n} \{ (I_{X^1_{lp^m}} - I_{X^2_{lp^m}})^2 + (Q_{X^1_{lp^m}} - Q_{X^2_{lp^m}})^2 \} \Big] \end{aligned} \tag{2.22}$$

Similarly, good codes correspond to the values of a_1, a_2^m and a_3^m, with $1 \le m \le n$, that result in the best distance spectrum.

It is known that CCs with good distance spectrum lead to TCs with good asymptotic performance, and CCs with the worst distance spectrum result in TCs with better performance at low SNR in the detriment of the good asymptotic performance. To this end, and depending on the target error rate, using NB-CCs with the best distance Euclidean distance spectrum does not necessary lead to the best NB-TCs. Therefore, later in this chapter, we consider NB-CCs with different distance spectrum characteristics.

2.4.2 DC sequence enumeration and truncated distance spectrum calculation

This section describes the algorithm we have developed to compute the first terms of the CSED spectrum of the NB-CC codes. In order to simplify the description of the search algorithm, only CCs with one parity polynomial are considered in this section ($n = 1$), but the described method can directly be generalized to codes with $n > 1$.

According to the previous section, in order to calculate the CSED spectrum of the code, one should be able to enumerate all possible sequences and to calculate the distance between each pair of DC sequences according to Eq. (2.21). In principle, the number of sequences to be considered grows exponentially with the considered number of trellis stages. However, since the trellises of the adopted code structure are fully connected,

some DC sequences can be very short: any pair of paths in the trellis diverging from a state at time i can converge to any state at time $i+2$, i.e., the shortest DC sequences have length 2.

In practice, it turns out that the enumeration of short DC sequences is sufficient to evaluate the lowest distances of the spectrum: we have observed that, whatever the values of coefficients a_1, a_2 and a_3, enumerating the length-2 and length-3 DC pairs of paths (see Fig. 2.10) is enough to find all the sequences corresponding to the minimum CSED d_1^2 and to the second minimum CSED d_2^2 of the code. This is guaranteed since, when considering length-3 sequences diverging from a state but not converging to another (i.e. truncated DC-4 sequences in Fig. 2.10), the obtained cumulated distances are greater than d_1 and d_2. This observation makes the process of evaluating the truncated minimum distance spectrum much easier.

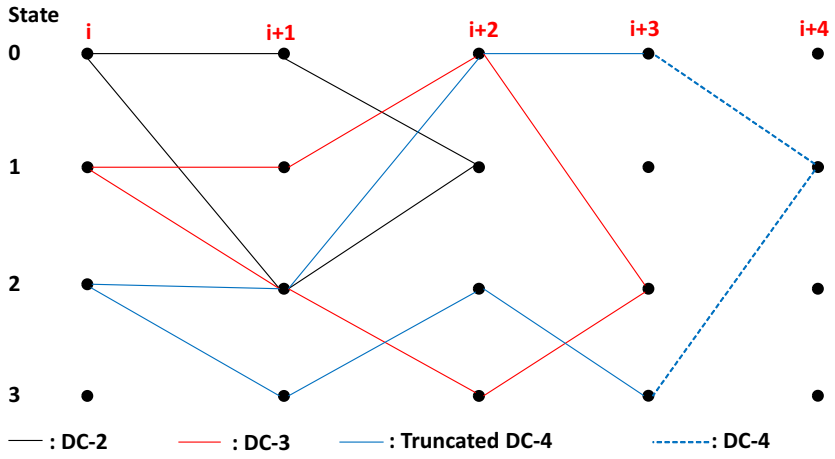

Figure 2.10: Examples of length-2, length-3, truncated length-4 and length-4 DC pairs of sequences in GF(4).

Therefore, for each examined code, we have enumerated length-2 and length-3 DC sequences, computed the corresponding CSED and stored the multiplicities for the first two distance terms d_1 and d_2 according to Algorithm 1. In the algorithm, E_i, s_i and p_i are the current state, the input value and the resulting parity at time i; $Dist(S1)(S2)$ represents the squared Euclidean distance between symbols $S1$ and $S2$ in the corresponding q-QAM constellation, d denotes the CSED between two competing DC sequences, and $Mult(d)$ represents the number of DC pairs having a CSED equal to d.

2.4.3 Discussion on the impact of the constellation mapping

As described earlier, a NB-CC is defined by the coefficients constituting its generator polynomials, a_1, a_2 and a_3. For a given value of a_1, a_2 and a_3, the distance spectrum of the coded modulation depends directly on the mapping of the encoded symbols X in GF(q) to the q-ary constellation C. A question that arises is whether changing the mapping has an impact on the distance spectrum of the best convolutional code that can be found by varying the coefficients a_1, a_2 and a_3 values. To begin the discussion, a total of $q!$ different mappings μ can be defined for a q-ary constellation C, here a q-QAM constellation. The effect on length-2 and length-3 DC sequences of varying the mapping

Algorithm 1: Calculation of the CSED spectrum of a NB-CC defined over GF(q).

Distance calculation of length-2 DC sequences:
for $E_i = 0$ *to* $q-1$ **do**
  for $E_{i+2} = 0$ *to* $q-1$ **do**
    for $s_i = 0$ *to* $q-1$ **do**
      Obtain E_{i+1} from Eq. (2.14).
      Calculate p_i according to Eq. (2.16).
      $s_{i+1} = E_{i+2} - a_1 E_{i+1}$.
      Calculate p_{i+1} according to Eq.(2.16).
      Save corresponding DC sequence $\{s_i, p_i, s_{i+1}, p_{i+1}\}$ in vector $DCseq$.
    end
    for $i = 0$ *to end of* $DCseq$ *vector* **do**
      for $j = i+1$ *to end of* $DCseq$ **do**
        $d = \sum_{k=0}^{3} Dist(DCseq(i)(k))(DCseq(j)(k))$.
        $Mult(d) = Mult(d) + 1$
      end
    end
  end
end
Distance calculation of length-3 DC sequences:
for $E_i = 0$ *to* $q-1$ **do**
  for $E_{i+3} = 0$ *to* $q-1$ **do**
    for $s_i = 0$ *to* $q-1$ **do**
      Calculate p_i according to Eq. (2.16).
      Obtain E_{i+1} from Eq. (2.14).
      for $s_{i+1} = 0$ *to* $q-1$ **do**
        Calculate p_{i+1} according to Eq. (2.16).
        Obtain E_{i+2} from Eq. (2.14).
        $s_{i+2} = E_{i+3} - a_1 E_{i+2}$.
        Obtain p_{i+2} from Eq. (2.16).
        Save corresponding DC sequence $\{s_i, p_i, s_{i+1}, p_{i+1}, s_{i+2}, p_{i+2}\}$ in vector $DCseq$.
      end
    end
    for $i = 0$ *to end of* $DCseq$ *vector* **do**
      for $j = i+1$ *to end of* $DCseq$ **do**
        $d = \sum_{k=0}^{5} Dist(DCseq(i)(k))(DCseq(j)(k))$.
        $Mult(d) = Mult(d) + 1$
      end
    end
  end
end

μ should be studied. If no effect resulting from the change of mapping μ is found, one can fix μ first and define the NB-CCs afterwards. Otherwise, a mutual optimization of the mapping and the convolutional code should be conducted. Two cases are considered for this study:

- The code is defined and kept unchanged (i.e. constant values are taken for a_1, a_2 and a_3 parameters), while the constellation mapping μ spans the different $q!$ possibilities,
- The mapping μ is kept unchanged, while each a_1, a_2 and a_3 code parameter spans the $q-1$ possible values.

If both cases provide two coded modulations with the same distance spectrum, we can then conclude that the mapping has no impact on the result of the search procedure.

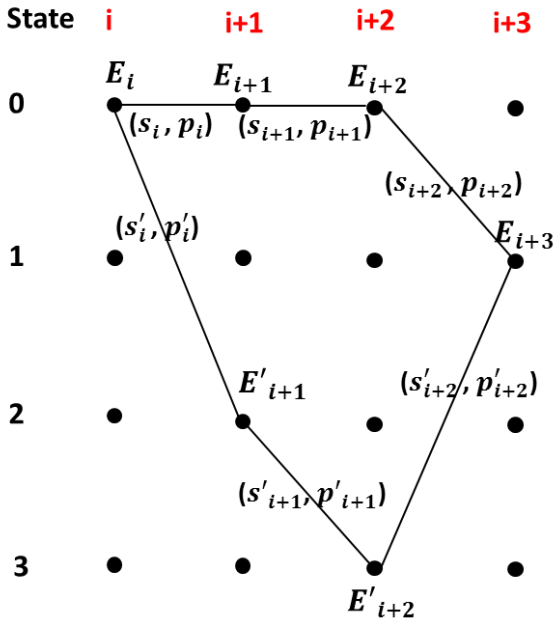

Figure 2.11: An example of length-3 DC sequences in a GF(4) code trellis.

Fig. 2.11 shows an example of two DC sequences of length 3. The exploration of length-2 and length-3 DC sequences can be divided into three steps at most:

- The first step focuses on the divergence section of the sequences, i.e. the section where they first diverge from the same state E_i to two different states E_{i+1} and E'_{i+1}.
- The second step addresses the trellis section (or sections for longer sequences) where these sequences do not have any state in common. This step corresponds to the transitions from E_{i+1} to E_{i+2} and from E'_{i+1} to E'_{i+2} in the example of Fig. 2.11. Note that for length-2 DC sequences, this step does not exist.
- The final step concerns the trellis section where the sequences converge to the same state E_{i+3}.

For the two previously mentioned study cases, we count the different possible values that the CSED between length-2 and 3 pairs of DC sequences can take. This evaluation is

done in the general context of an arbitrarily shaped q-ary constellation, which represents the worst case scenario. When QAM constellations are used, the number of distinct Euclidean distances decreases, due to the symmetry imposed by the structure of a QAM constellation.

2.4.3.1 Case 1 – parameters of the code are constant and μ changes

Let us denote by N_μ the total number of different possible values of the CSED between any pair of DC sequences having length up to 3.

- Step 1: This step corresponds to a divergence step. Here, Euclidean distances are computed between sequences emanating from the same state. Each of the corresponding transitions is labeled by a systematic and a parity symbol. The contribution of this step to the overall CSED is the sum of the Euclidean distance between the systematic values, and the Euclidean distance between the parity values according to Eq. (2.21). The maximum number of distinct distances between each pair of symbols is the number of 2-element combinations in a set of q elements, denoted by $\binom{q}{2}$. When varying the mapping, any combination of two possible values of each symbol type is possible. Therefore, the contribution of step 1 to N_μ is $\binom{q}{2}^2$.
- Step 2: In this intermediate step, Euclidean distances are computed between transitions without any state in common. Here, a pair of competing sequences can share the same systematic symbol ($s_{i+1} = s'_{i+1}$) or parity symbol ($p_{i+1} = p'_{i+1}$), which adds one possibility for the distance with respect to step 1. However, it can never share the same systematic and parity symbols at the same time, due to the code structure, following Eq. (2.14) and (2.16). Therefore, the contribution of step 2 to N_μ is $\left(\binom{q}{2} + 1\right)^2 - 1$.
- Step 3: Dictated by the code structure, by exploiting the existing symmetry with regards to step 1, we can deduce that the contribution of step 3 to N_μ is also $\binom{q}{2}^2$.

In summary, $N_\mu = \left(\left(\binom{q}{2} + 1\right)^2 - 1\right) \cdot \binom{q}{2}^4$.

2.4.3.2 Case 2 – μ is constant and the parameters of the code change

When the mapping is fixed, the total number of different possible CSED values, between any pair of DC sequences having length up to 3, is denoted by N_a.

- Step 1: When emanating from the same state, the transitions are labeled by all possible systematic symbols. This corresponds to $\binom{q}{2}$ different possible sub-terms in the Euclidean distance. When varying a_1, a_2 and a_3, if all combinations of q^2 parities labeling any couple of transitions are spanned, then the ensemble of generated codes covers all possible parity symbol combinations. This can be verified by taking one reference transition (systematic and parity values kept constant) and by validating that the second transition can span all possible parity values, even when the systematic part is constant. In Fig. 2.11, for the same starting state E_i, if transition (s_i, p_i) is taken as a reference, then parity p'_i should be able to span

the $q-1$ possible values (excluding $p_i = p'_i$) when s'_i is kept unchanged. From Eq. (2.14) and (2.16), we have:

$$p_i + p'_i = a_2(s_i + s'_i) \tag{2.23}$$

When a_2 spans GF(q), $p_i + p'_i$ takes the $q-1$ different non zero values. Therefore, p'_i can actually take any possible value in $\{\mathrm{GF}(q)\backslash\{p_i\}\}$. Consequently, the contribution of step 1 to N_a is $\binom{q}{2}^2$.

- Step 2: With code structure S_2, q^2 different competing transitions are considered in this step regardless of the code parameters a_1, a_2 and a_3, since any two transitions differ at least by their systematic or parity value. Therefore, the contribution of step 2 to N_a is $\binom{q^2}{2}$.
- Step 3: Again, dictated by the code structure, by exploiting the existing symmetry with regards to step 1, the contribution of step 3 to N_a is $\binom{q}{2}^2$.

In summary, $N_a = \binom{q^2}{2} \cdot \binom{q}{2}^4$.

When comparing the total number of distinct Euclidean distances resulting from the two proposed cases, one can conclude about the role played by the mapping in the definition of the best NB-CC distance spectrum. First, $\binom{q}{2}^4$ is a common factor of N_μ and N_a. Therefore, the same number of distance values are considered in steps 1 and 3 with both approaches. It is then sufficient to compare the terms $\binom{q^2}{2}$ and $\left(\binom{q}{2}+1\right)^2 - 1$:

$$\begin{aligned} \delta_N &= \binom{q^2}{2} - \left(\binom{q}{2}+1\right)^2 + 1 \\ &= \frac{q^2(q^2-1)}{2} - \frac{q(q-1)(q^2-q+4)}{4} \\ &= \frac{q(q-1)^2(q+4)}{4} > 0 \;\forall\; q > 0 \end{aligned} \tag{2.24}$$

Eventually, since $\delta_N > 0$, then $N_a > N_\mu$. Hence, when the mapping μ is kept unchanged, more distances are considered than when the code is kept unchanged. In conclusion, the distance spectrum of a NB-CC given by DC sequences of length 2 or 3 does not depend on the mapping μ. The latter can be kept constant, and the variation of the code parameters (a_1, a_2 and a_3) is sufficient in order to define the best code using the proposed search procedure. Therefore, in the rest of this document, the mapping function μ is kept constant and is given in Table 2.3 for 4-QAM, 16-QAM and 64-QAM constellations. It was chosen such that the binary image of the constellation symbols follows a Gray mapping. The binary images of the symbols are denoted by b_1b_0, $b_3b_2b_1b_0$ and $b_5b_4b_3b_2b_1b_0$ for the 4-QAM, 16-QAM and 64-QAM, respectively, with the highest index corresponding to the most significant bit of the symbol representation.

2.4.4 Search results for $R = 1/2$ NB-CCs over GF(q) combined with q-QAM, q= 4, 16, 64 and simulation results

We have applied the methodology described in the previous section to the case of rate-1/2 NB-CCs defined in GF(4), GF(16) and GF(64) and mapped to 4-QAM, 16-QAM and 64-QAM constellations. The primitive polynomials used to generate the elements of

4-QAM			
Q value	b_1	I value	b_0
+1	0	+1	0
-1	1	-1	1
16-QAM			
Q value	b_3b_1	I value	b_2b_0
+3	00	+3	00
+1	01	+1	01
-1	11	-1	11
-3	10	-3	10
64-QAM			
Q value	$b_5b_3b_1$	I value	$b_4b_2b_0$
+ 7	000	+ 7	000
+5	001	+5	001
+3	011	+3	011
+1	010	+1	010
-1	110	-1	110
-3	111	-3	111
-5	101	-5	101
-7	100	-7	100

Table 2.3: Binary mapping of the in-phase I and quadrature Q axes for 4-, 16- and 64-QAM.

the Galois fields are: $P_{\mathrm{GF}(4)}(x) = 1 + x + x^2$, $P_{\mathrm{GF}(16)}(x) = 1 + x^3 + x^4$ and $P_{\mathrm{GF}(64)}(x) = 1 + x^2 + x^3 + x^5 + x^6$.

For each field order q with $q = 2^m$, several codes were found with the same truncated distance spectra. We have selected three specific codes for each value of q: C_1 denotes a code showing the worst distance spectrum in GF(q), i.e. with the lowest minimum CSED, C_3 is one the codes that enjoy the best distance spectrum, and code C_2 was selected as a code with a "medium" distance spectrum. Table 2.4 provides the values of coefficients a_1, a_2 and a_3 for these three specific codes resulting from the search in GF(4), GF(16) and in GF(64). The first two minimum CSEDs, d_1 and d_2, were calculated by enumerating the corresponding length-2 and length-3 DC sequences as explained in Section 2.4.2. Table 2.4 also provides the corresponding multiplicities $n(d_1)$ and $n(d_2)$, i.e. the number of DC sequences with distances d_1 and d_2.

GF(4)			
Code	C_1	C_2	C_3
(a_1, a_2, a_3)	$(3,1,0)$	$(1,0,3)$	$(3,2,2)$
d_1^2 $(= d_{min}^2)$	4	6	10
$n(d_1)$	8	48	96
d_2^2	6	8	12
$n(d_2)$	40	68	100
GF(16)			
Code	C_1	C_2	C_3
(a_1, a_2, a_3)	$(12,4,0)$	$(10,12,3)$	$(13,7,11)$
d_1^2 $(= d_{min}^2)$	1.20	2.00	4.00
$n(d_1)$	22128	5532	22484
d_2^2	1.60	2.40	4.80
$n(d_2)$	16596	8424	141144
GF(64)			
Code	C_1	C_2	C_3
(a_1, a_2, a_3)	$(41,2,0)$	$(41,1,24)$	$(31,5,18)$
d_1^2 $(= d_{min}^2)$	0.38	1.14	1.52
$n(d_1)$	238422	1542390	652698
d_2^2	0.57	1.23	1.61
$n(d_2)$	230886	4111444	1084014

Table 2.4: Three representative codes obtained from the search over GF(4), GF(16) and GF(64), with the first two terms of the CSED spectra d_1^2 and d_2^2 and the corresponding multiplicities $n(d_1)$ and $n(d_2)$.

The error rate performance of the selected codes was assessed through Monte Carlo simulations over a Gaussian channel. We simulated the transmission of blocks of 100 symbols, corresponding to 200 bits in the case of GF(4), 400 bits in the case of GF(16) and 600 bits in the case of GF(64). The symbol error rate curves are shown in Fig. 2.12 for the codes in GF(4), in Fig. 2.13 for the codes in GF(16), and in Fig. 2.14 for the codes in GF(64). These figures show the range of performance that can be obtained with the proposed code structure and search process. In the presented graphs, codes denoted as C_3

show the best performance in terms of Symbol Error Rate (SER), and codes denoted as C_1 show the worst performance. The observed results confirm that the proposed criterion in the search procedure is accurate, and it can be adopted as the only criterion to define NB-CCs over GF(q). Although the performance curves have not been reported in this manuscript, we have verified that codes sharing the same distance spectra show the same performance curves.

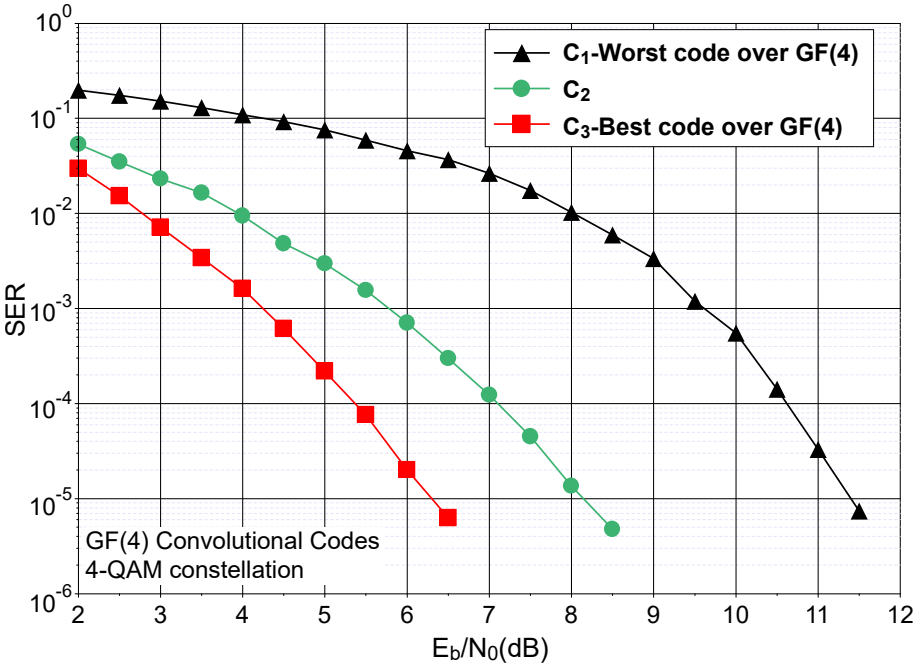

Figure 2.12: Performance comparison of rate-1/2 convolutional codes over GF(4) in terms of symbol error rates in AWGN channel.

We have made some comparisons with NB-CCs previously published in the literature. For instance, we have compared our best code C_3 defined over GF(64) with the best code over ring $\mathbb{Z}_{64}$ mapped to 64-QAM proposed in [81]. Fig. 2.15 shows the comparison in terms of bit error rate. This comparison shows that our best code in GF(64) outperforms the best proposed code over $\mathbb{Z}_{64}$ by around 0.5 dB.

In addition, we have also compared this code C_3 with the best known 64-state binary CC with the same decoding complexity, mapped to a 64-QAM constellation: the convolutional code, with generator polynomials $\left(1, \frac{171}{133}\right)$ in octal, is known to be the 64-state binary recursive systematic convolutional code with the highest minimum Hamming distance [87]. Bit Error Rate (BER) curves are plotted in Fig 2.16. It is shown that code C_3 designed over GF(64) and mapped to a 64-QAM constellation, outperforms the binary code. This gain is in the order of 0.7 dB, which is in accordance with the gain predicted by the capacity comparison, between a coded modulation and a bit-interleaved coded modulation for a 64-QAM constellation and at spectral efficiency corresponding to a coding rate 1/2 (see Fig. 1.7).

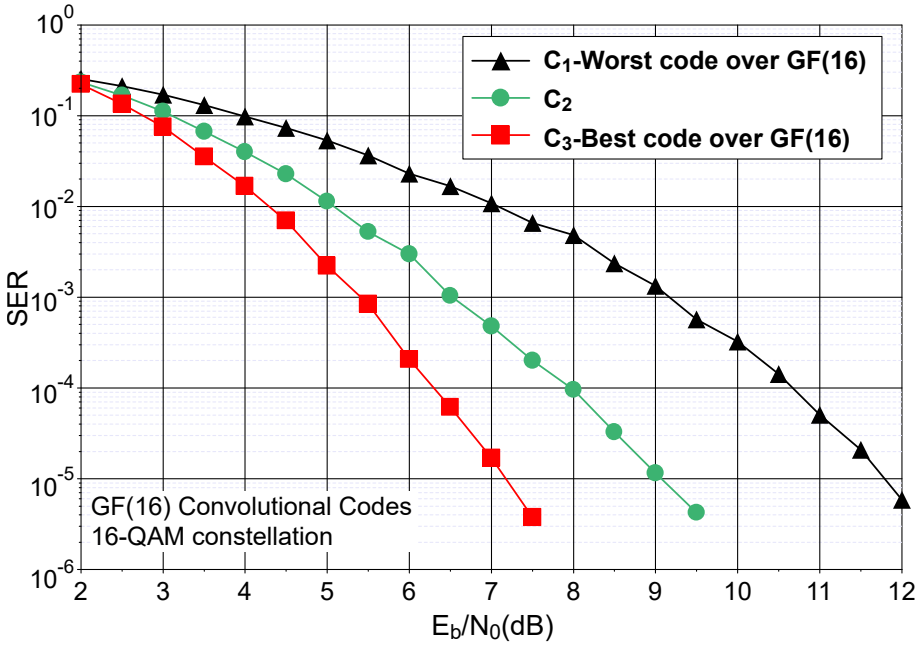

Figure 2.13: Performance comparison of rate-1/2 convolutional codes over GF(16) in terms of symbol error rates in AWGN channel.

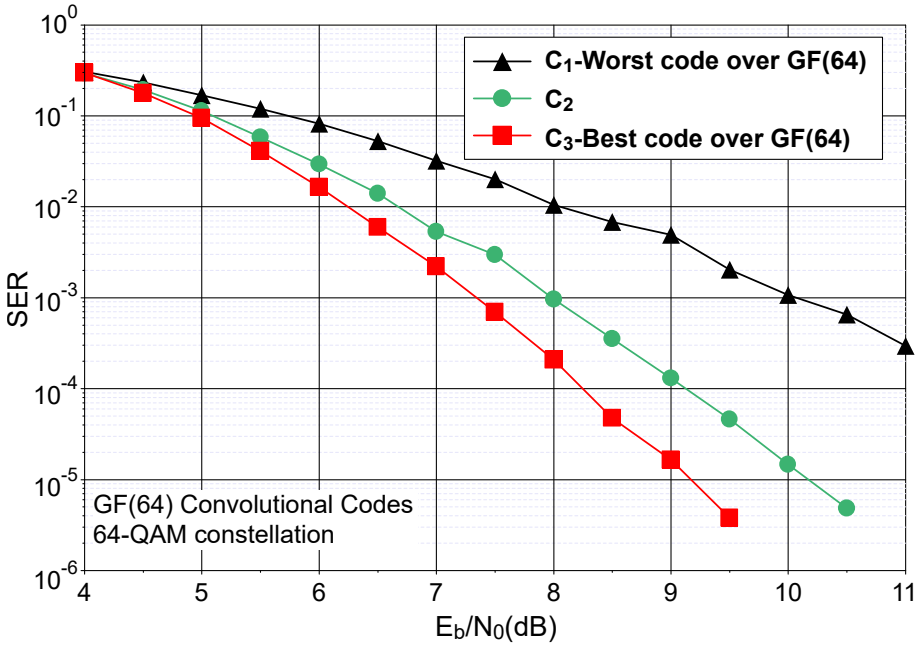

Figure 2.14: Performance comparison of rate-1/2 convolutional codes over GF(64) in terms of symbol error rates in AWGN channel.

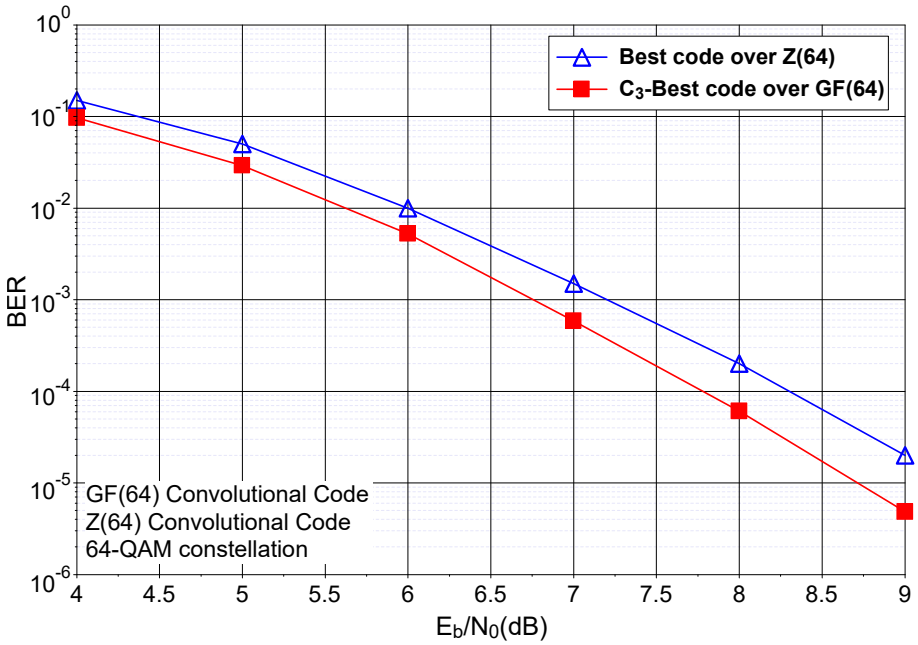

Figure 2.15: Performance comparison of the proposed rate-1/2 convolutional code defined over GF(64) (code C_3) with the code defined over $\mathbb{Z}_{64}$ and labeled C_1 in [81], in terms of bit error rate. Transmission over AWGN channel using 64-QAM constellation.

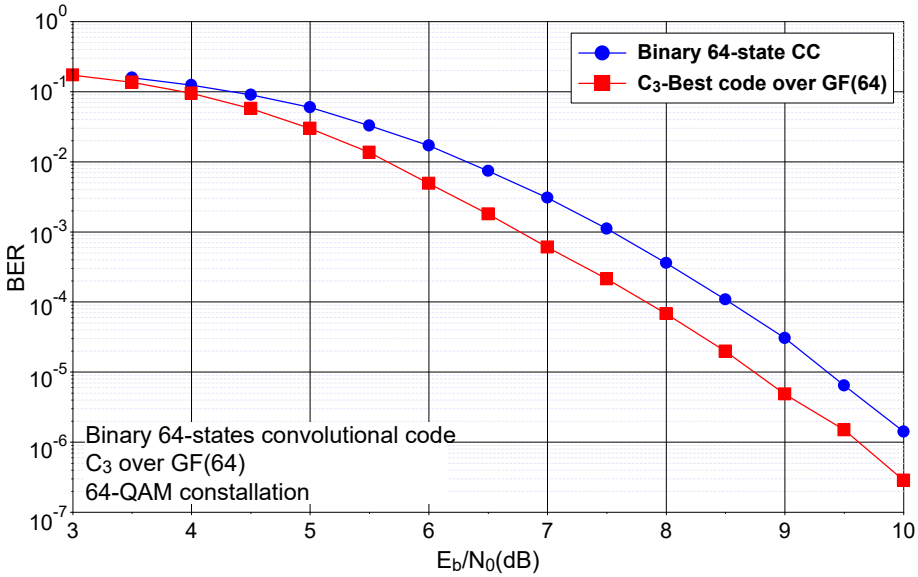

Figure 2.16: Performance comparison of the proposed rate-1/2 convolutional codes over GF(64) with the best known 64-state binary code in terms of bit error rate. Transmission over AWGN channel, using a 64-QAM constellation.

2.4.5 Application to NB-CCs over GF(q) with code rates $R < 1/2$.

When the number of parity symbols n is greater than 1 (see Fig. 2.9), coding rates $R < 1/2$ can be achieved. Using the same search procedure as described above, NB-CCs can be defined for different rates over different finite fields.

We have run a search, using Eq. (2.22) for the CSED calculation between DC sequences, for $n = 2$, 3 and 4 to find a code with the best truncated distance spectrum (denoted by C_5 for $n = 2$, C_7 for $n = 3$ and C_9 for $n = 4$) and a code with the worst truncated distance spectrum (denoted by C_4 for $n = 2$, C_6 for $n = 3$ and C_8 for $n = 4$), for each value of n and each field order. We adopted a recursive search procedure: starting from the $R = 1/2$ codes ($n = 1$) reported in Table 2.4, the search procedure was launched for the selection of an additional parity, resulting in $R = 1/3$ codes ($n = 2$). Then, the same procedure was iterated to build the $R = 1/(n + 1)$ codes from the $R = 1/n$ ones, for $n = 3$ and 4. Table 2.5 shows an instance of the best and worst codes found for different considered values of n and q.

The codes defined in Table 2.5 were assessed through SER performance evaluation. Fig. 2.17, 2.18 and 2.19 show the performance comparison of codes C_4 and C_5 for coding rate $R = 1/3$, C_6 and C_7 for $R = 1/4$, C_8 and C_9 for $R = 1/5$ over GF(4), GF(16) and GF(64) respectively. In the shown results, the designed code with the best distance spectrum outperforms the one with the worst distance spectrum in all scenarios, which further justifies the design criterion.

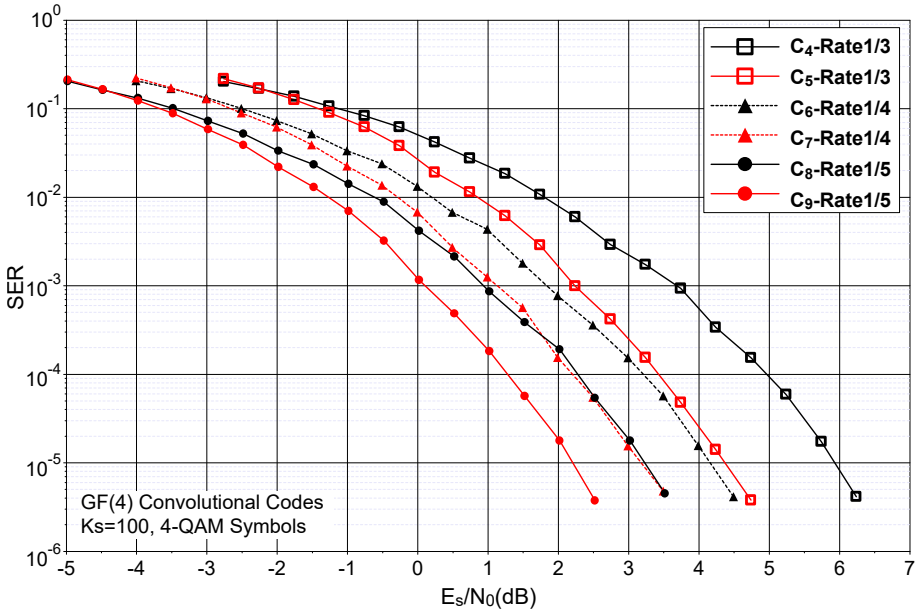

Figure 2.17: Performance comparison of the proposed NB-CCs over GF(4), with different numbers of parities resulting in different coding rates. Transmission over an AWGN channel using a 4-QAM constellation.

GF(4)						
Rate	$R = 1/3$		$R = 1/4$		$R = 1/5$	
Code	C_4	C_5	C_6	C_7	C_8	C_9
(a_1, a_2^1, a_3^1)	(1,0,1)	(3,2,2)	(1,0,1)	(3,2,2)	(1,0,1)	(3,2,2)
(a_2^2, a_3^2)	(2,2)	(3,1)	(1,1)	(3,1)	(0,2)	(3,1)
(a_2^3, a_3^3)			(2,3)	(1,2)	(1,2)	(1,2)
(a_2^4, a_3^4)					(3,3)	(1,3)
d_1^2 $(= d_{min}^2)$	10	16	16	20	20	26
$n(d_1)$	48	144	96	48	96	96
d_2^2	12	20	20	22	24	28
$n(d_2)$	20	88	40	96	48	48
GF(16)						
Rate	$R = 1/3$		$R = 1/4$		$R = 1/5$	
Code	C_4	C_5	C_6	C_7	C_8	C_9
(a_1, a_2^1, a_3^1)	(1,0,5)	(13,7,11)	(5,1,1)	(13,7,11)	(7,0,1)	(13,7,11)
(a_2^2, a_3^2)	(11,15)	(3,4)	(1,2)	(3,4)	(0,3)	(3,4)
(a_2^3, a_3^3)			(7,14)	(8,7)	(8,15)	(8,7)
(a_2^4, a_3^4)					(5,10)	(5,12)
d_1^2 $(= d_{min}^2)$	4.8	6.4	6.8	9.2	9.2	12
$n(d_1)$	30588	7910	732	17691	17414	19065
d_2^2	5.2	6.8	7.2	10	10	13.2
$n(d_2)$	5856	11364	11796	22708	47954	6607
GF(64)						
Rate	$R = 1/3$		$R = 1/4$		$R = 1/5$	
Code	C_4	C_5	C_6	C_7	C_8	C_9
(a_1, a_2^1, a_3^1)	(13,35,52)	(31,5,18)	(27,10,43)	(31,5,18)	(53,61,24)	(31,5,18)
(a_2^2, a_3^2)	(0,7)	(22,55)	(22,0)	(22,55)	(0,7)	(22,55)
(a_2^3, a_3^3)			(0,4)	(16,7)	(44,0)	(16,7)
(a_2^4, a_3^4)					(0,12)	(15,12)
d_1^2 $(= d_{min}^2)$	1.52	2.95	2.57	4.66	2.28	6.57
$n(d_1)$	476844	45164	22076	3731	6965	108254
d_2^2	1.61	3.23	2.76	5.14	3.04	7.23
$n(d_2)$	1882	675846	14896	353905	44142	257821

Table 2.5: Two representative codes obtained from the search over GF(4), GF(16) and GF(64), with the first two terms of the CSED spectra d_1^2 and d_2^2 and the corresponding multiplicities $n(d_1)$ and $n(d_2)$. C_4, C_6 and C_8 are code instances with the worst distance spectra in each Galois field, while C_5, C_7 and C_9 are the code instances with the best distance spectra. Different coding rates are achieved: $R = 1/3$, $R = 1/4$ and $R = 1/5$.

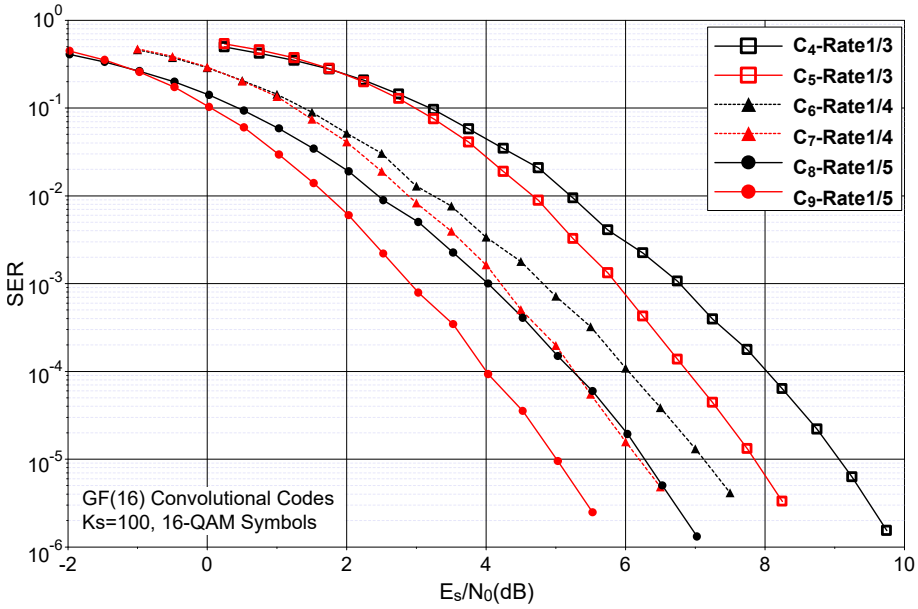

Figure 2.18: Performance comparison of the proposed NB-CCs over GF(16), with different numbers of parities resulting in different coding rates. Transmission over an AWGN channel using a 16-QAM constellation.

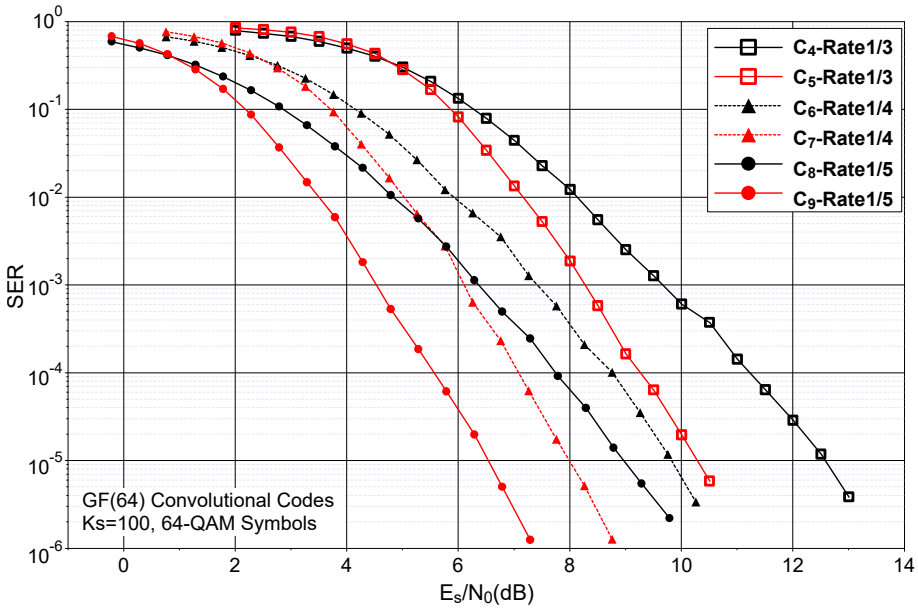

Figure 2.19: Performance comparison of the proposed NB-CCs over GF(64), with different numbers of parities resulting in different coding rates. Transmission over an AWGN channel using a 64-QAM constellation.

2.5 Constellation downsizing technique

As shown in Eq. (2.21), the CSED between two sequences depends on the number of different symbols between the sequences and on the distance between these symbols in the constellation. Therefore, for a given Galois field order, the shape of the constellation can be an additional degree of freedom for the design of the coded modulation scheme. In this section, we consider the design of NB-CCs in the specific context of *constellation downsizing*. A downsized constellation is obtained by merging several symbols of the original q-QAM constellation into a single symbol. As shown later in this section, downsizing can improve the correction performance of the coded modulation while reducing the demodulation complexity. Due to the square shape of the q-QAM constellations under study, we consider merging the symbols four by four in order to downsize the q-QAM constellation to a $q/4$-QAM constellation. The downsizing process is illustrated in Fig. 2.20 for a 64-QAM constellation.

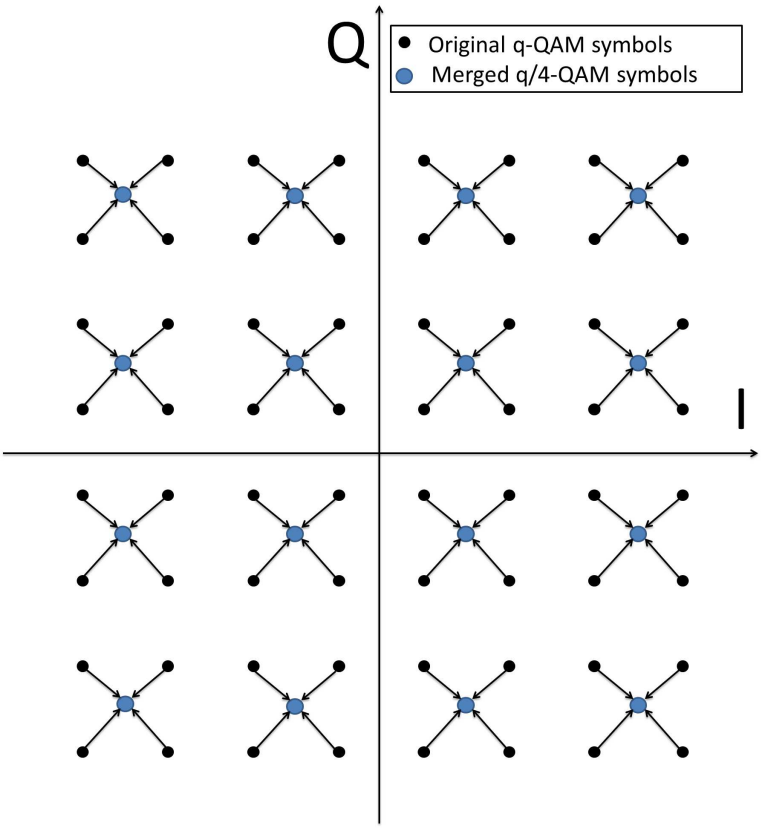

Figure 2.20: Constellation downsizing from 64-QAM to 16-QAM.

The merged symbols can no longer be distinguished in the constellation and the corresponding terms in Eq. (2.21) are zero. Therefore, the number of non-zero terms in Eq. (2.21) is decreased, but the minimum Euclidean distance between the symbols in the new $q/4$-QAM constellation is increased, provided that the average energy per symbol is kept constant through the merging process. The minimum distance between two symbols of a conventional q-QAM is

$$d_{min}(q\text{-QAM}) = \sqrt{\frac{6E_s(q\text{-QAM})}{q-1}} \tag{2.25}$$

$E_s(q\text{-QAM})$ is the average energy per symbol:

$$E_s(q\text{-QAM}) = \sqrt{\frac{\sum_{j=1}^{q}(I_j^2 + Q_j^2)}{q}} \tag{2.26}$$

where I_j and Q_j denote the in-phase and quadrature coordinates of the constellation symbols. Since the average energy is the same for both constellations ($E_s(q\text{-QAM}) = E_s(\frac{q}{4}\text{-QAM})$), we can write from Eq. (2.25): $d_{min}(q\text{-QAM})^2.(q-1)/6 = d_{min}(\frac{q}{4}\text{-QAM})^2.(q/4-1)/6$. Therefore, the ratio between the minimum Euclidean distances of the original and downsized constellations is equal to:

$$\frac{d_{min}(\frac{q}{4}\text{-QAM})}{d_{min}(q\text{-QAM})} = \sqrt{\frac{q-1}{q/4-1}} \tag{2.27}$$

Eq. (2.27) provides the increase factor of the minimum distance resulting from the downsizing. For a 64-QAM constellation downsized to 16-QAM,

$$\frac{d_{min}(16\text{-QAM})}{d_{min}(64\text{-QAM})} = \sqrt{\frac{21}{5}} \tag{2.28}$$

and for a 16-QAM constellation downsized to 4-QAM:

$$\frac{d_{min}(4\text{-QAM})}{d_{min}(16\text{-QAM})} = \sqrt{5} \tag{2.29}$$

A main advantage of using the constellation downsizing technique is that the receiver does not need to implement a q-QAM demodulator but a $q/4$-QAM demodulator, where only $q/4$ distances should be computed for each received symbol instead of q. The demodulator complexity is therefore reduced. As a consequence, when downsizing is applied, for each received signal, the corresponding decoder receives $q/4$ likelihood values from the demapper, representing the probabilities for the received symbol to be equal to a symbol from the $q/4$ clusters. The decoder then replicates the received $q/4$ likelihood values to form q likelihood values at the input of the decoder. The ambiguities will be removed by the decoding process.

Constellation downsizing can be particularly efficient if the associated code is designed in such a way to limit the occurrence of two merged symbols in any two competing DC sequences as much as possible. To this end, a second search round for NB convolutional codes was carried out according to the search methodology described in Section 2.4.2. The aim is still to maximize the minimum CSED between two DC sequences, taking into account the merging constraint, which reduces the number of non-zero terms in Eq. (2.21) but increases the minimum distance between symbols in the downsized constellation.

In this context, NB-CCs with one parity ($n = 1$) and rate $R = 1/2$ are considered. An exhaustive search for the values of a_1, a_2 and a_3 was carried out to find the best code for the underlying merged constellation.

Alternatively, we also considered a simpler design method relying on the best code found for the classical 64-QAM constellation. In fact, since downsizing applies merging of the four closest constellation symbols, we can still rely on the previous exhaustive search results with a slight modification. Hence, we proposed to apply a new coefficient a_s to

the systematic symbol before its transmission, in order to avoid as much as possible that the systematic symbols labeling competing transitions in DC sequences are merged in the downsized constellation when the corresponding parity symbols are merged[1]. Fig. 2.21 shows the modified code structure, with the additional coefficient a_s.

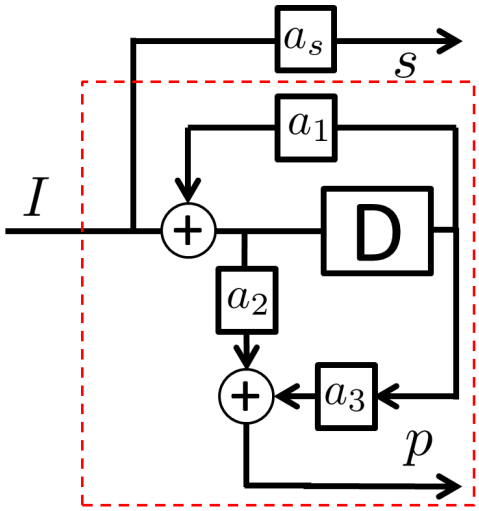

Figure 2.21: NB-CC structure, according to S_2 from Fig. 2.7, introducing the new coefficient a_s.

Starting from the codes C_3 of Table 2.4, we ran a search on coefficient a_s only, in order to find the value yielding the best code for the downsized constellation. It turned out that the same distance spectrum was obtained as the one yielded by the exhaustive search for the three coefficients a_1, a_2 and a_3 with the downsized constellation.

The main advantage of the proposed approach relies in the limited number of required modifications to introduce constellation downsizing in an existing system.

In addition to the reduced demapping complexity, the proposed downsizing technique provided codes with increased CSED. Table 2.6 compares the truncated distance spectra of the code C_3 designed over GF(64) and mapped to a 64-QAM with the new code C_{10} optimized to be mapped onto a downsized 16-QAM constellation.

We can observe that C_{10} shows a better distance spectrum than C_3. The SER performance of C_3 and C_{10} is evaluated and compared in Fig. 2.22. At low SNR values, C_3 has a slightly better performance than C_{10}, which is the result of the ambiguity between the merged symbols in GF(64) for code C_{10}. When a frame is erroneous, C_{10} shows a high number of symbol errors, since the decoder is not able to recover the transmitted codeword in the trellis due to the ambiguity imposed by symbol merging. When SER$\leq 10^{-2}$, C_{10} begins to outperform C_3, since the decoder of C_{10} can better remove the ambiguities between merged symbols.

Frame Error Rate (FER) performance of C_3 and C_{10} is compared in Fig. 2.23. The SNR region where C_3 outperforms C_{10} (in terms of SER) does not make any difference at the FER level, since the FER is very close to one in this region. When the SNR grows beyond 5.5 dB, the FER drops below 1 for both codes and C_{10} begins to outperform C_3, with a gain of up to 0.5 dB for FER$\leq 10^{-2}$.

[1]Firstly, we also added an additional coefficient a_p to the parity output. However, it appeared that no additional increase in the minimum distance was provided when using two additional coefficients instead of one. Therefore, we finally only kept coefficient a_s.

Code	C_3	C_{10}
(a_1, a_2, a_3, a_s)	(31,5,18,1)	(31,5,18,7)
Constellation	64-QAM	16-QAM
d_1^2 $(=d_{min}^2)$	1.52381	1.6
$n(d_1)$	652698	924464
d_2^2	1.61905	2.0
$n(d_2)$	1084014	13232944

Table 2.6: Comparison of the distance spectra between two GF(64) codes optimized for a mapping onto a 64-QAM constellation and a downsized 16-QAM constellation, respectively.

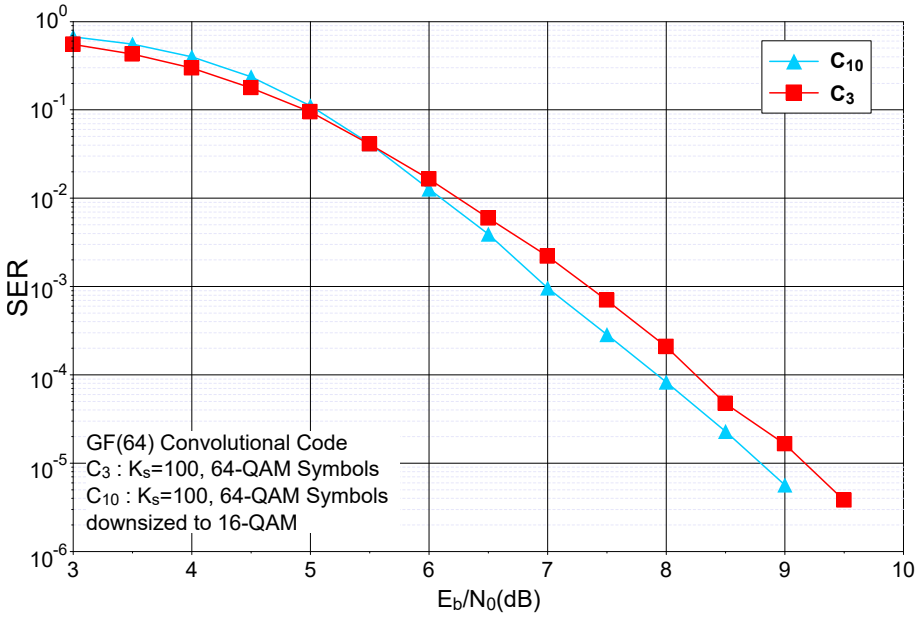

Figure 2.22: SER performance comparison of a rate-1/2 NB-CC over GF(64), before and after constellation downsizing.

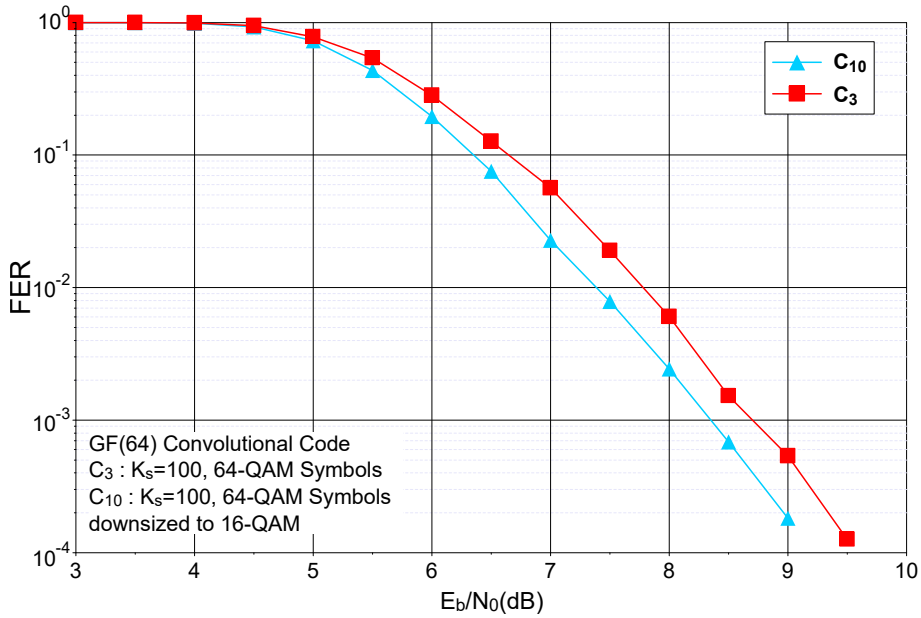

Figure 2.23: FER performance comparison of a rate-1/2 NB-CC over GF(64), before and after constellation downsizing.

2.6 Optimizing the NB-CC coefficients for binary modulations

We complete this chapter with the study of the proposed NB codes over GF(64) when a Binary Phase-Shift Keying (BPSK) constellation is used for signal transmission. These codes are compared with the best known 64-state RSC binary CC. In this context, a modification is applied to the system model where the symbols are converted to the binary form to constitute the binary image of the NB code before being mapped to the BPSK constellation. Here, NB-CCs with one parity ($n = 1$) and rate $R = 1/2$ are considered. To obtain an appropriate code for this configuration, the modified structure of NB-CCs is used, where a_s is taken into consideration, as described in Fig. 2.21. Therefore, code C_3 of Table 2.4 was used and we searched the value of a_s yielding the maximum Hamming distance of the resulting code, C_{11}, when mapped to a BPSK constellation.

On the other hand, a new search was conducted over GF(64), to find the new values of a_1, a_2 and a_3 defining the code with the best Hamming distance when used with a BPSK constellation (code C_{12}). Note that, as far as code C_{12} is considered, when optimizing parameters a_1, a_2 and a_3 in the presence of BPSK mapping, the best code yielded $a_s = 1$, which means that the optimization of this additional parameter can be dropped in these conditions. The binary images of both NB-CCs over GF(64) are compared with the best known 64-state RSC binary CC in Fig. 2.24 in terms of BER performance and in Fig. 2.25 in terms of FER performance. The truncated distance spectra of the three codes are shown in Table 2.7. A better distance spectrum is observed for the new defined NB-CC C_{12}, with respect to the two other codes.

Fig. 2.24 and 2.25 show that the binary image of the non-binary code C_{11} performs close to the binary code in terms of BER and outperforms it in terms of FER. The binary image of the non-binary code C_{12} slightly outperforms both other codes, especially at high SNR, in terms of BER and FER.

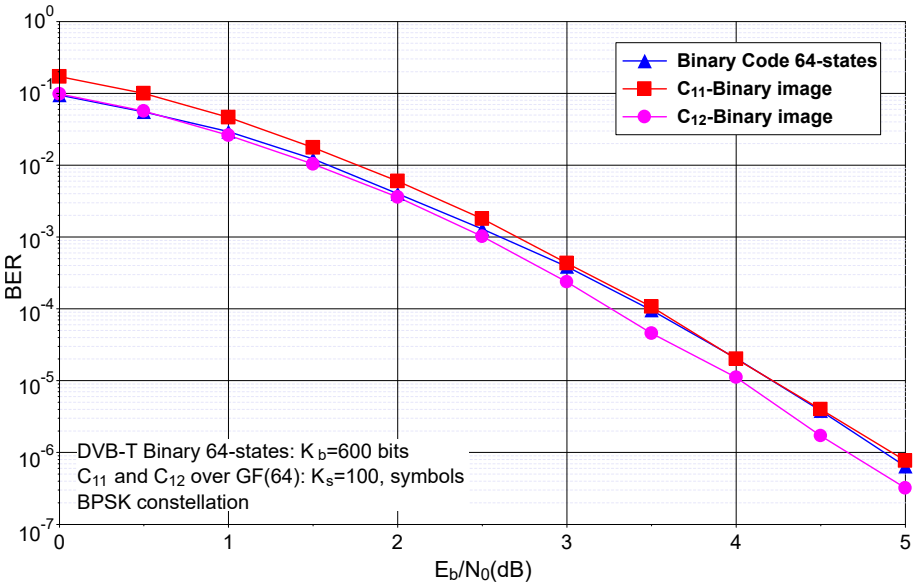

Figure 2.24: Performance comparison of rate-1/2 codes C_{11} and C_{12} defined over GF(64) with the $\left(1, \frac{171}{133}\right)$ binary 64-state CC, in terms of BER over the AWGN channel.

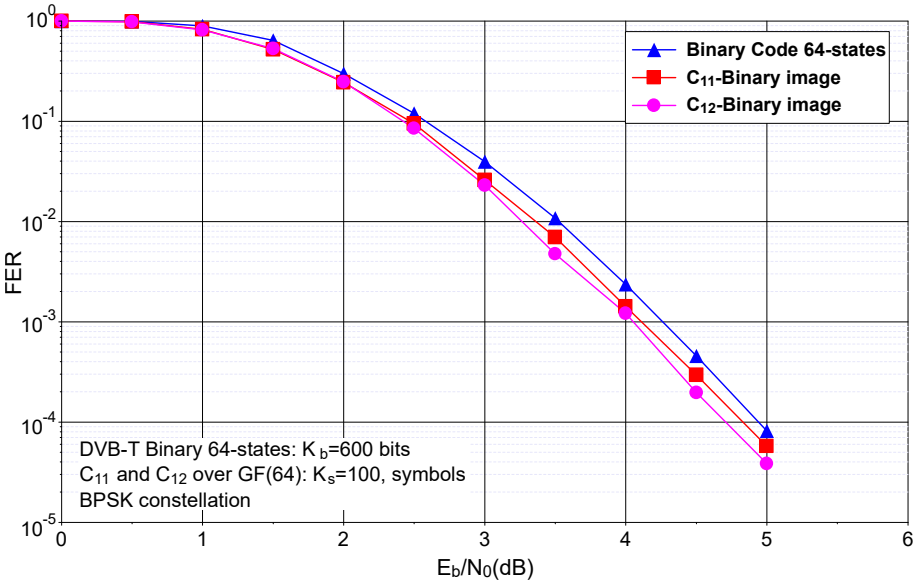

Figure 2.25: Performance comparison of rate-1/2 codes C_{11} and C_{12} defined over GF(64) with the $\left(1, \frac{171}{133}\right)$ binary 64-state CC, in terms of FER over the AWGN channel.

Code	C_{11}	C_{12}	Binary Code
Generators	(a_1, a_2^1, a_3^1, a_s) $(31,5,18,22)$	(a_1, a_2^1, a_3^1, a_s) $(26,40,51,1)$	133-171
d_1^2 $(=d_{min}^2)$	9	10	10
$n(d_1)$	6	22	11
d_2^2	10	12	12
$n(d_2)$	15	23	38

Table 2.7: Truncated Hamming distance spectra of the two rate-1/2 codes defined over GF(64), mapped onto a BPSK constellation, and of the best binary 64-state code.

This result is somewhat surprising, specially for C_{11} since one can observe from Table 2.7 that the binary code has a better Hamming distance spectrum than the NB codes when a BPSK constellation is used for transmission. We have not fully analyzed this behavior so far, but we believe that this is due to the NB decoding algorithm, which is more robust since it operates in the symbol domain.

2.7 Conclusion

This chapter deals with the design of RSC codes over GF(q). A new code structure is investigated, based on an updated structure of an accumulator. This structure has shown to offer a good trade-off between complexity and error correction performance.

The optimization of the cumulated Euclidean distance spectrum is the criterion used in order to define the best NB-CC over each finite field. In addition, a low-complexity method is proposed to calculate the distance spectrum of such codes, since the enumeration of the pairs of length-2 and length-3 DC sequences is sufficient for the calculation of the first two distance terms of the spectrum.

Also, the proposed structure allows the search space for the code to span all the combinations of systematic and parity symbols for each transition between two states in the trellis. It was also shown that the distance properties of the resulting codes are independent of the constellation mapping.

The designed codes offer better performance than the non-binary codes previously proposed in the literature. They also outperform their binary counterparts when combined with their corresponding QAM modulation or with lower order modulations. This study can be considered as a first step to design NB-TCs. The exploration of interleaving techniques, puncturing patterns and the reduction of the decoding complexity are the next steps to complete this work.

Note that the construction of NB-CCs over GF(q) proposed in this chapter was published and presented at the IEEE Global Communications Conference (GLOBECOM18):

R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah, "Design of Low-Complexity Convolutional Codes over GF(q)", IEEE Global Commun. Conf.: Communication Theory (Globecom2018 CT), Abu Dhabi, United Arab Emirates, Dec. 2018.

Another paper is being prepared, targeting the techniques of constellation downsizing and the optimization of NB-CCs for binary constellations.

Chapter 3

Towards the design of non-binary turbo codes: interleaving, puncturing, union bound

After having designed NB constituent CCs with good distance properties, the next step towards the construction of good NB-TCs defined over GF(q) involves designing interleavers and puncturing patterns that provide high coding gains in all SNR regions. To the best of our knowledge, such a study is still missing from the literature. This chapter intends to address this topic by investigating the design of efficient interleavers and puncturing patterns for NB-TCs. It also deals with the computation of the union bounds for NB-TCs in order to predict their asymptotic behavior.

This chapter is organized as follows: in Section 3.1, we first analyze the validity of the well-known design constraints of interleavers for binary TCs in the case of NB-TCs. Moreover, in Section 3.1.4, new additional constraints targeting the mitigation of correlation in the iterative (turbo) decoding process are proposed. Section 3.2 is then devoted to the study of puncturing for NB-TCs. Two types of puncturing are investigated: NB symbol puncturing and bit puncturing. A method is then proposed in Section 3.3 to estimate the Euclidean distance spectrum of NB-TCs in order to compute their truncated union bound. Finally, Section 3.4 concludes the chapter.

3.1 Study of the influence of different interleaving techniques on NB-TCs

Similarly to binary TCs, the error correction performance of NB-TCs, especially the error floor, highly depends on the applied permutation. Different interleaving constraints were defined in the case of binary TCs, which were shown to have an impact on their asymptotic performance. A simple first approach consists in evaluating the application of the well-known binary interleaver constraints to the NB-TCs. To this end, a simple comparison of the resulting performance with/without each identified constraint can be performed to assess its impact.

3.1.1 Random interleaver

Random interleavers are generally used to assess the performance of the coding structure regardless of the effect of interleaving. By generating a random permutation for each transmitted frame, sufficiently long Monte Carlo simulations provide the average performance of all interleavers [56]. Moreover, by definition, such interleavers do not incorporate any type of design constraints.

To evaluate the corresponding performance, simulations are performed for a NB-TC defined over GF(64) and associated with a 64-QAM constellation. No puncturing is applied, resulting in coding rate $R = 1/3$. Two NB-TCs are considered, the first one is constructed from the worst NB-CC over GF(64) in terms of ED spectrum (code C_1 in Table 2.4), whereas the second NB-TC uses the best NB-CC over GF(64) (code C_3 in Table 2.4). Corresponding results are provided in Fig. 3.1. High error floors are obtained: neither TC is able to achieve frame error rates lower than 10^{-3}.

By studying the error patterns obtained in this case, we observed that, since the interleaver has no spread constraint, it commonly happens that two neighboring symbols in the non-interleaved frame are still neighbors in the interleaved frame (see for example Fig. 3.2). Due to the adopted CC structure (see Fig. 2.7), this induces length-2 DC sequences with the same low CSED for both constituent codes. Under such conditions, it is highly likely that erroneous sequences in one decoder cannot be corrected by the other decoder, thus causing the high error floor.

In order to remedy this problem, S-random interleavers were applied to increase the minimum spreading value, which avoids the occurrence of such error patterns.

3.1.2 S-random interleaver

When considering the use of S-random interleavers [51], a new permutation is generated for each transmitted frame, respecting a minimum spreading value. Fig. 3.3 shows that S-random interleaving leads to lower error floors for the resulting NB-TC, compared to the previous interleaving model. The NB-TCs using constituent codes C_1 and C_3 can now achieve 10^{-3} and $\sim 10^{-6}$ of FER in the error floor, respectively.

Moreover, in order to assess the sensitivity of the performance of NB-TCs to the spreading value, random interleavers with largely different spread values are compared. Fig. 3.4 compares the FER curves of two NB-TCs using constituent code C_3 and using S-random interleavers of length $K_s = 900$ with minimum spreading values $S_{min} = 30$ and $S_{min} = 8$. Note that the maximum achievable spreading value for $K_s = 900$ is $\sqrt{2K_s}$,

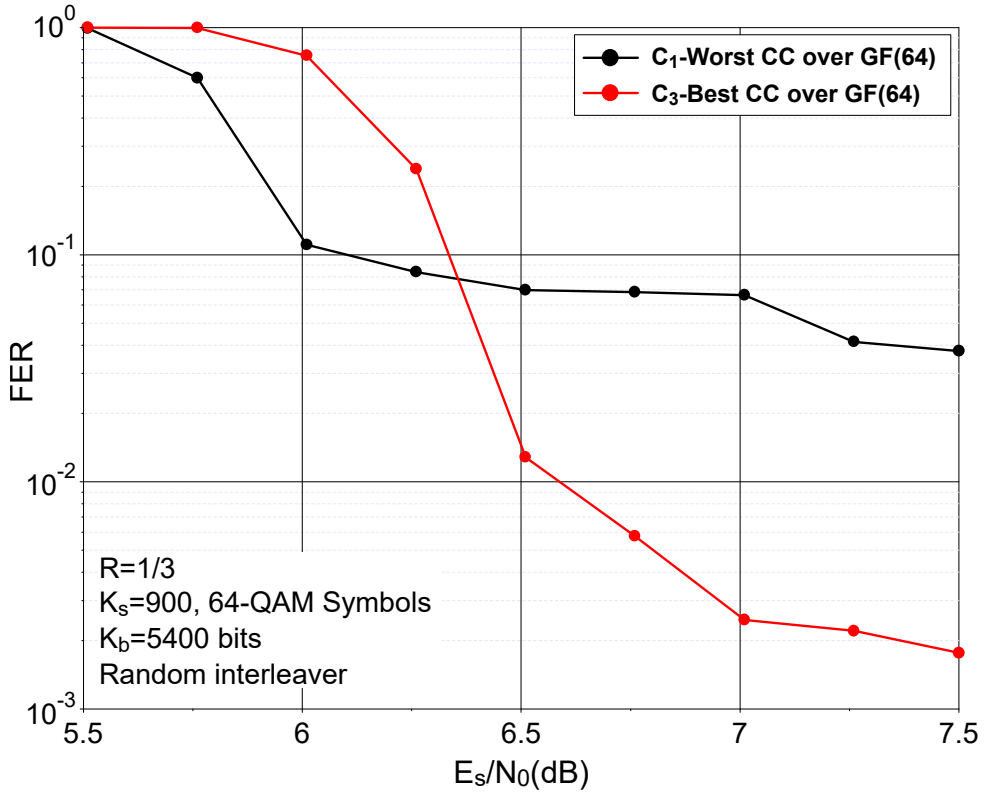

Figure 3.1: Frame error rate performance of NB-TCs defined over GF(64) and using random interleavers. Transmission over AWGN channel using 64-QAM modulation. Frame size $K_s = 900$ symbols, coding rate $R = 1/3$.

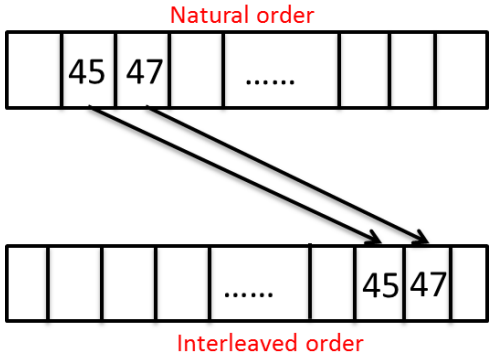

Figure 3.2: Error pattern due to the low spread value, here $S=2$, in a random interleaving graph.

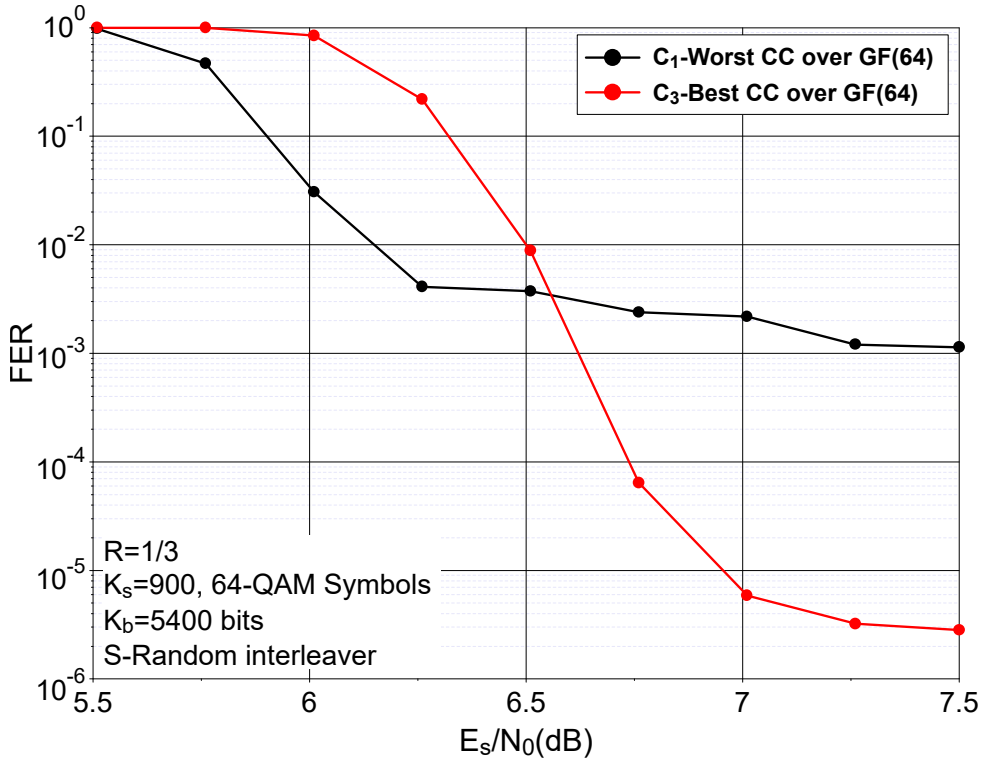

Figure 3.3: Frame error rate performance of NB-TCs defined over GF(64) and using S-random interleavers. Transmission over AWGN channel using 64-QAM modulation. Frame size $K_s = 900$ symbols, coding rate $R = 1/3$.

i. e. 42. This comparison shows a quasi identical performance of these two codes. In fact, in the case of NB-TCs with constituent encoders having fully connected trellises, approaching the maximum achievable spread value does not impact the error floor. This is due to the fact that, for the considered component codes, the smallest CSED values are due to very short (typically length 2 and 3) DC sequences. Therefore, spread values as low as 8 are sufficient to avoid low CSED due to the close vicinity of symbols in both the natural and interleaved order. Targeting spread values close to the maximum achievable value may actually be counterproductive since this puts hard constraints on the other design criteria of the interleaver.

Interestingly, since S-random interleavers do not consider the length of correlation cycles, the remaining low-CSED error patterns are due to composite DC sequences with short correlation cycles. Fig. 3.5 shows an example of sequence with girth $g = 4$. The minimum CSED value between this sequence and the closest competing sequence is then equal to 4 times the minimum CSED of the component code. Moreover, low girth values entails high correlation in the decoding process. To achieve higher minimum CSED values and to limit the correlation in the decoder, structured interleavers with higher girth value have to be implemented, as explained in Section 3.1.3.

3.1.3 ARP interleaver

To address both problems of low spread and low correlation girth, the Almost Regular Permutation (ARP) [60, 54] model is investigated for NB-TCs. The ARP model is quite general since it was shown to encompass the well-known Quadratic Permutation Poly-

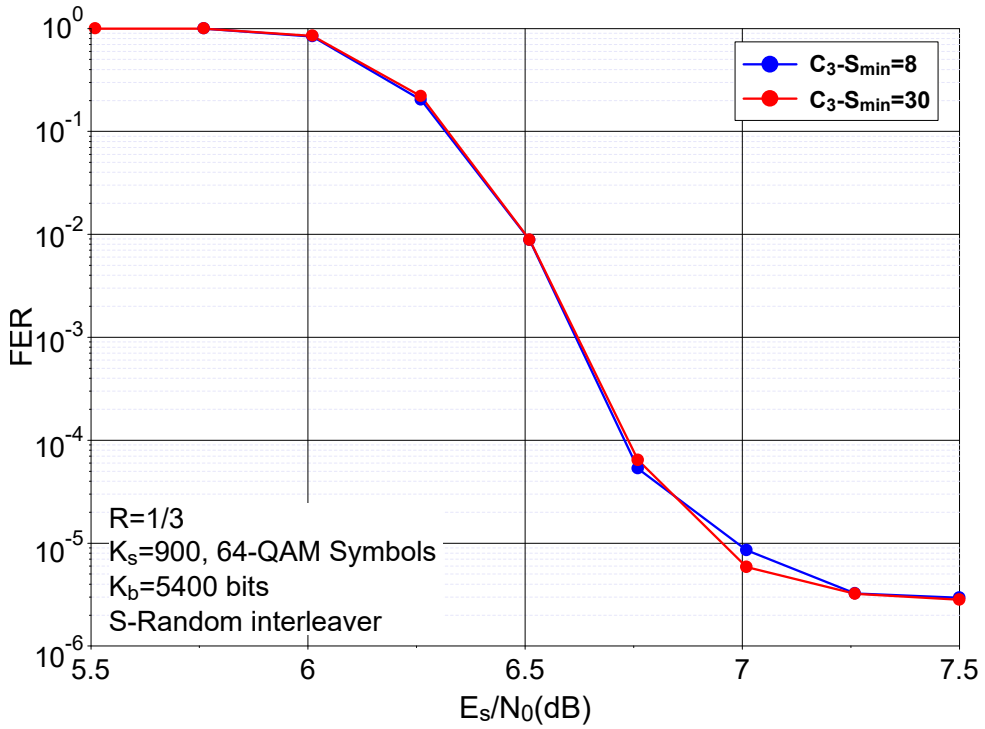

Figure 3.4: Frame error rate performance comparison between two NB-TCs based on constituent code C_3 defined over GF(64) and using different S-random interleavers. Transmission in AWGN channel with 64-QAM modulation. K_s = 900 symbols and R = 1/3.

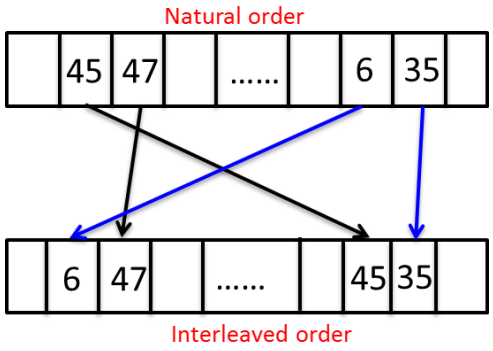

Figure 3.5: Error patterns due to a low girth value, here $g = 4$, in the S-random interleaving graph.

nomial (QPP) and Dithered Relative Prime (DRP) models [12]. In [13], a method is proposed to design ARP interleavers with high spread and girth values, while trying to maximize the minimum Hamming distance of the TC.

The adopted ARP interleaver follows the model given in [12]:

$$\Pi(i) = (P.i + S(i \bmod Q)) \bmod K \tag{3.1}$$

where $i = 0...K-1$ stands for the interleaved address of the data symbol and $\Pi(i)$ its corresponding address in the natural order. P stands for the period of the corresponding regular interleaver, Q is the disorder degree, and S defines the vector of the Q shift values.

An ARP interleaver can be seen as the superposition of Q regular interleavers with period QP and with different shift offsets. Q has to be chosen as a divider of the information frame size. For the design of NB-TCs over GF(64), we carried out a search for ARP parameters with $Q = 4$ for information blocks with $K_s = 900$ symbols. For these parameters, the maximum obtained girth value is $g = 8$ and, amongst the high number of possible interleavers, we chose one yielding the highest spreading value, i. e. 30. The parameters of the adopted ARP interleaver are given in Table 3.1. Note that several other interleavers with the same features can be found and used instead.

S_{min}	Girth	P	$S(0)$	$S(1)$	$S(2)$	$S(3)$
30	8	137	0	854	396	362

Table 3.1: ARP interleaver for NB-TC over GF(64), $K_s = 900$ symbols.

The FER performance of rate-1/3 NB-TCs using constituent codes C_1 and C_3 with the interleaver parameters given in Table 3.1 is depicted in Fig. 3.6. The error floors of both codes, especially the one based on C_3, are lowered with respect to those obtained when randomly generated interleavers are considered (Fig. 3.1 and Fig. 3.3). Consequently, the ARP interleaver appears to be a suitable interleaving model for NB-TCs. Indeed, with this design we can reach low frame error rates with constituent code C_3.

Nevertheless, in order to still achieve lower error floors, we have investigated the error patterns in the error floor when ARP interleavers are adopted. Fig. 3.7 shows an example of observed error pattern in the error floor region. In this example, the minimum correlation girth $g = 8$ is respected but the error results from the rather limited extrinsic information exchange in the considered correlation cycle. This is due to the fact that the interleaver maps three short DC sequences into the same DC sequences in the interleaved order. Under such a scenario, when the first decoder fails to decode the three short sequences due to channel errors, it is very likely that the second decoder behaves similarly, since it has to decode the same transmitted symbols, as shown in the figure. We have observed that most of the errors occurring in the error floor with ARP interleavers were due to such patterns. Therefore, we have introduced new design criteria for the selection of the ARP interleaver parameters, that help mitigate these correlation issues. These new constraints are described in Section 3.1.4.

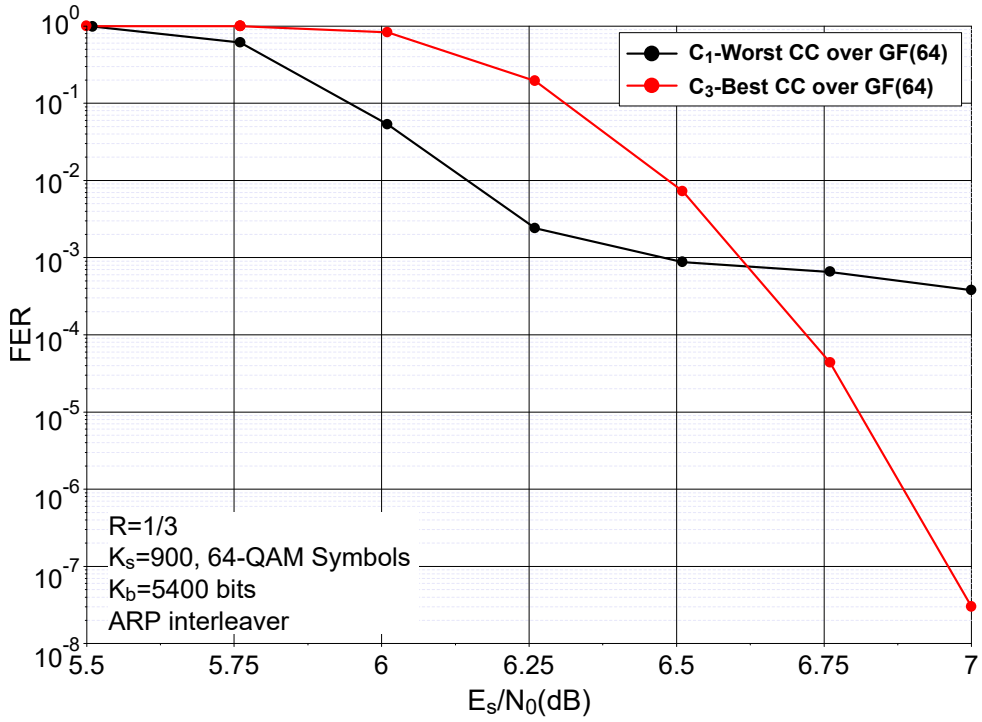

Figure 3.6: Frame error rate performance of NB-TCs defined over GF(64) and using ARP interleavers. Transmission over AWGN channel using 64-QAM modulation. Frame size $K_s = 900$ symbols, coding rate $R = 1/3$.

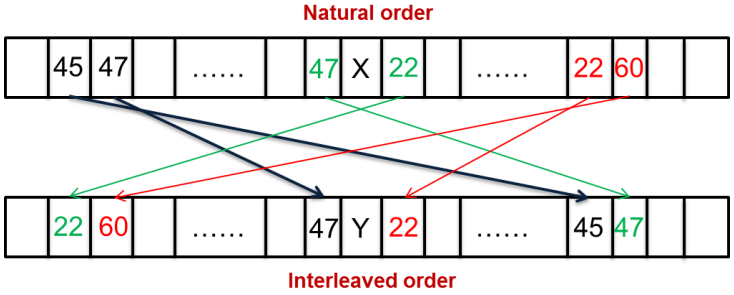

Figure 3.7: Example of possible low-CSED error pattern resulting from a long correlation cycle, here $g = 8$.

3.1.4 Reducing the correlation effect in NB-TCs using ARP interleavers

For short frame sizes and high coding rates, correlation in the decoding process was identified several times as one of the main factor in reducing the error correction capabilities of TCs [30, 53]. Some observations seem to support this statement: in some cases, TCs are not able to approach the asymptotic performance predicted by the calculation of the union bound from the estimated distance spectrum [13]. In Section 3.1.3, the ARP parameters were selected so as to maximize the correlation girth of the TC and to attempt to approach the maximum achievable girth value given by the Moore bound [88]. In this section, we propose additional constraints and criteria aiming at reducing the correlation in the decoding process.

3.1.4.1 Minimizing the multiplicity of length-g correlation cycles

The first measurable parameter that can be considered is the multiplicity of the correlation girth, $n(g)$, that is the total number of correlation cycles with minimum length g. For example, Fig. 3.8 shows a girth-6 correlation graph with 3 cycles having length $g = 6$. In order to reduce the correlation, the interleaver should be designed to minimize the multiplicity of the correlation girth g, $n(g)$:

$$\text{Select } (Q,P,S) \text{ to } \underset{\text{cycle length}=g}{\text{minimize}} \; n(g) \tag{3.2}$$

This criterion can be generalized to a larger set of correlation cycles with lengths greater than the girth.

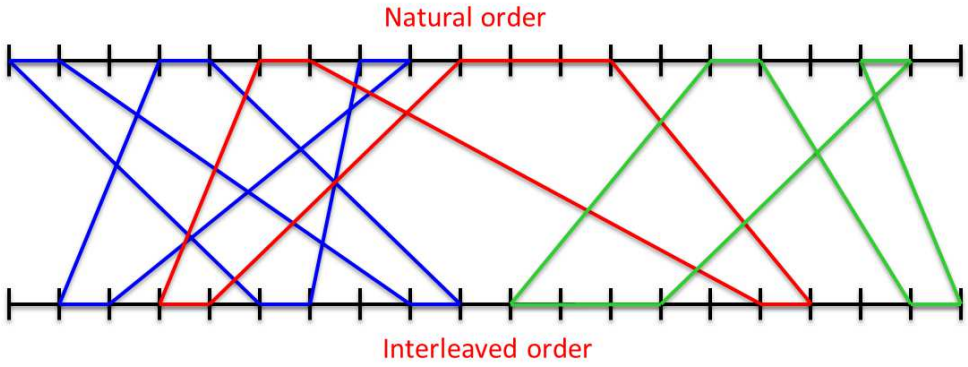

Figure 3.8: An example of a correlation graph for a TC with $g = 6$ and $n(g) = 3$.

3.1.4.2 Maximizing the information exchange between component decoders in the minimum correlation cycles

For a given correlation cycle length, we have observed that the structure of the cycles has an impact on the error correction capability of the decoder. Actually, the correlation in the decoding process can be better mitigated when each information in a short cycle can benefit from a large code diversity, i. e. when parity symbols provided by a high number of different and distant trellis sections participate in the cycle. In long contiguous trellis

sections, the correlation between the parity symbols is mostly caused by the convolutional component code and their contribution to the decoding process remains local.

For instance, Fig. 3.9 shows two length-7 cycles: in the red cycle, each symbol benefits from symbols coming from three other trellis sections whereas in the other case, each symbol benefits from symbols coming from five different and distant trellis sections. The code diversity is therefore higher in the latter case than in the former. This means that cycles with long contiguous trellis sections should be avoided. As for the spatial distance between the different contributing trellis sections, it is guaranteed to be large in practical applications, due to the spreading properties of the interleavers.

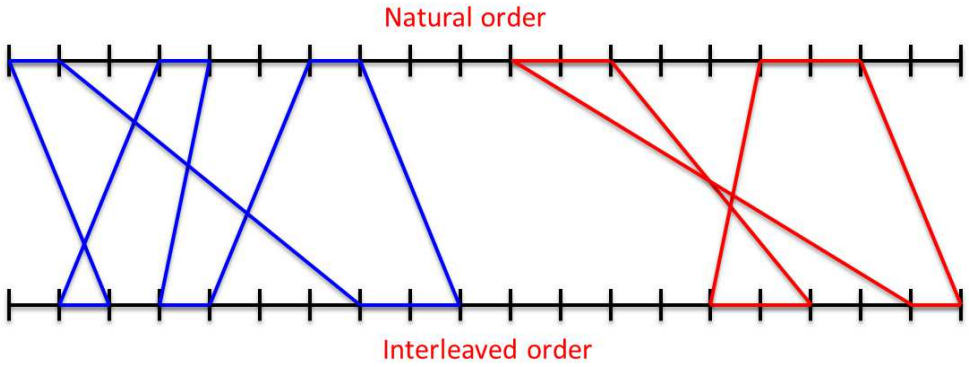

Figure 3.9: An example of a correlation graph for a TC with $g = 7$. $T_g = 6$ for the blue cycle, and $T_g = 4$ for the red one.

A proposed design criterion for the interleaver is the maximization of the number of non-contiguous trellis sections participating in short correlation cycles. Let $T_g(\mathbf{d},\mathbf{d}')$ be the number of transitions between $\mathbf{d}$ (the information frame in the natural order) and $\mathbf{d}'$ (the information frame in the interleaved order) in the correlation cycles of length g, the interleavers parameters are selected based on:

$$\text{Select } (Q,P,S) \text{ to } \underset{\text{cycle length}=g}{\text{maximize}} \; T_g(\mathbf{d},\mathbf{d}') \tag{3.3}$$

With high values of $T_g(\mathbf{d},\mathbf{d}')$, more decoding iterations are needed for a particular information symbol to get back to its original position through the information exchanges: autocorrelation is reduced.

The condition in Eq. (3.3) is of primary importance for the minimum length correlation cycles but can be generalized to a larger set of correlation cycles with short lengths.

3.1.4.3 Minimizing the multiplicity of symbol couples in minimum correlation cycles

The third criterion considered in this study is the number of times that a couple of symbols at positions i_1 and i_2, $i_1 \neq i_2$, participates in short correlation cycles. Let us denote it by $m(i_1,i_2)$ where $\{i_1,i_2\} \in [0,K-1]^2$, and K is the information frame length. Fig. 3.10 shows an example of a correlation graph where the couple of symbols (d_{i_1},d_{i_2}) belongs to two length-5 correlation cycles.

Given that symbols in short correlation cycles are prone to decoding errors, the multiplicity $m(i_1,i_2)$ in short cycles should be minimized to reduce the propagation of errors

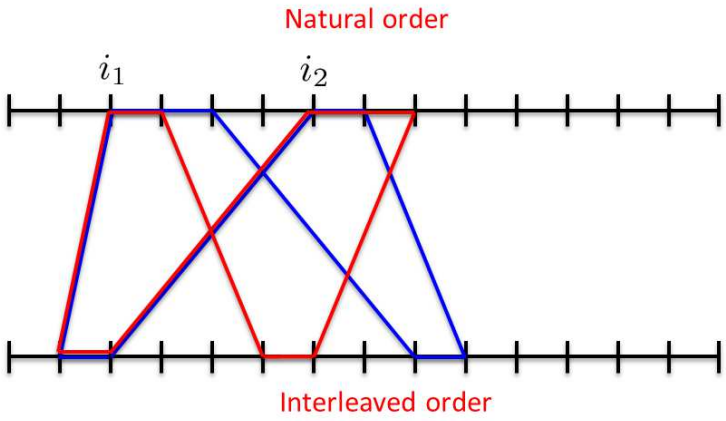

Figure 3.10: An example of a correlation graph for a TC with $g = 5$, and $m(i_1,i_2) = 2$.

throughout the decoding iterations:

$$\text{Select } (Q,P,S) \text{ to } \underset{\text{cycle length}=g}{\text{minimize}} \; m(i_1,i_2) \tag{3.4}$$

3.1.4.4 Numerical results

Correlation problems in turbo decoding can be primarily observed for short block sizes. Therefore, we apply the proposed criteria to design ARP interleavers for the encoding of information blocks consisting of $K_s = 30$ GF(64) symbols (equivalent to $K_b = 180$ bits). We adopt the following methodology: first, a set of ARP interleavers is generated according to minimum spread and correlation girth criteria, as described in [89]. A first set of interleavers is obtained with minimum spreading value $S_{min} = 5$ and girth $g = 4$. Then, a refinement phase is carried out, using the criteria presented in Sections 3.1.4.1, 3.1.4.2 and 3.1.4.3. Many interleavers result from this refinement phase, which are expected to provide similar error correction. An interleaver (ARP I) is picked up in the resulting subset while a second interleaver (ARP II) is selected for comparison purposes in the first set. ARP II satisfies the same spread and girth constraints as ARP I, but has not undergone the refinement phase and has different $n(g)$ and $m(i_1,i_2)$ values. Table 3.2 provides the interleaver parameters for ARP I and ARP II as well as the values of $n(g)$, $T_g(\mathbf{d},\mathbf{d}')$ and $m(i_1,i_2)$.

ARP	Q	P	$(S(0),...,S(Q-1))$	$n(g)$	$T_g(\mathbf{d},\mathbf{d}')$	$m(i_1,i_2)$
I	3	19	(0,25,14)	2	4	1
II	5	7	(0,17,3,22,28)	4	4	2

Table 3.2: Parameters Q, P and S of ARP I and ARP II interleavers for $K_s = 30$ symbols (eq. to $K_b = 180$ bits). Minimum spread $S_{\min} = 5$, girth $g = 4$, corresponding multiplicity $n(g)$, minimum number of $(\mathbf{d},\mathbf{d}')$ transitions in a length-4 cycle, $T_g(\mathbf{d},\mathbf{d}')$, and multiplicity of symbol couples $m(i_1,i_2)$ for $g = 4$.

The FER performance of the two resulting NB-TCs is evaluated by simulation over AWGN channel with 4-QAM modulation. At the decoder side, 8 iterations are performed,

using the scaled Max-Log-MAP algorithm. The best NB-CC over GF(64) in terms of CSED, C_3, is used as a constituent encoder for the designed NB-TCs. Fig. 3.11 compares the impact of the interleaver design on the FER performance of the resulting NB-TC for coding rate $R=1/3$. Thanks to the correlation reduction, ARP I outperforms ARP II by 0.2 dB at FER= 10^{-8}.

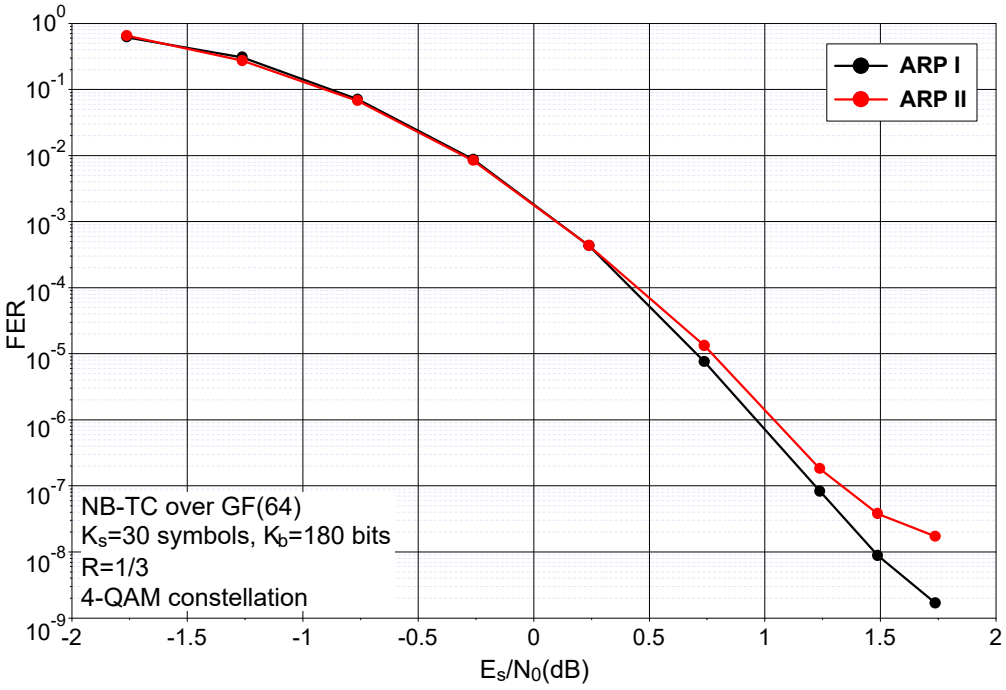

Figure 3.11: FER performance of NB-TCs defined over GF(64) for K_s = 30 symbols (K_b = 180 bits). ARP interleavers defined in Table 3.2. AWGN channel with 8 decoding iterations of the NB Max-Log-MAP algorithm. $R=1/3$ and 4-QAM modulation.

3.2 Puncturing NB-TCs

When one parity symbol is generated by each constituent encoder, TCs operate at the mother coding rate $R=1/3$. When high-throughput communications are targeted, higher coding rates are generally required. To achieve such rates, puncturing is applied [11] by disregarding some parity and/or systematic bits/symbols generated by the TC encoder.

For binary TCs, different puncturing methods can be applied. Lately, a joint optimization of the puncturing patterns and interleaver parameters was proposed in [13]. Thanks to this method, TCs largely improve their error correcting performance, especially at high coding rates.

For NB-TCs, two types of puncturing can be imagined: symbol puncturing and bit puncturing. Symbol puncturing involves disregarding some symbols at the output of the NB-TC encoder, while bit puncturing consists in removing some bits in the binary image of the coded symbols and reconstructing new symbols with the remaining bits.

3.2.1 Symbol puncturing

Short correlation cycles are particularly affected by puncturing, since removing systematic and/or parity symbols participating to a cycle in the correlation graph tends to lower the code diversity. This naturally leads to higher correlation in the decoding process. Let np_s and np_p be the number of non-punctured systematic and parity symbols in a short correlation cycle, respectively. In Fig. 3.12 an example of a short correlation cycle is shown, with $np_s = 4$ and $np_p = 5$. When np_s and np_p take low values, the code diversity experienced by data and parity symbols in these cycles is naturally reduced. In order to mitigate this detrimental effect, we designed the puncturing patterns so as to maximize the values of np_s and np_p in short correlation cycles.

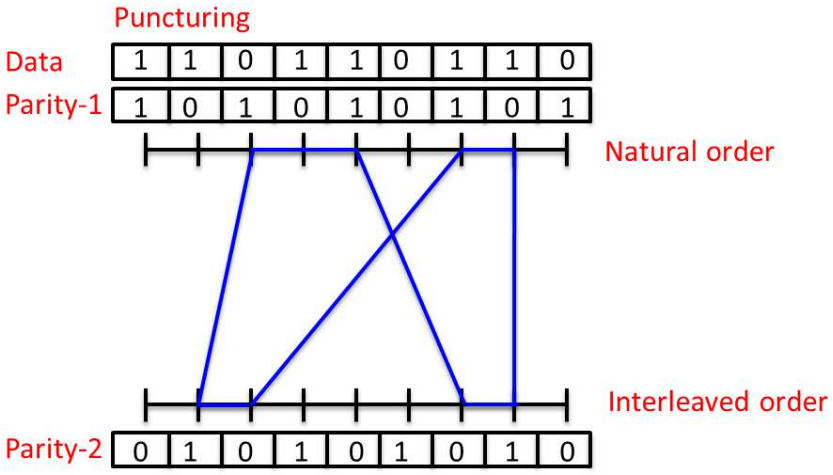

Figure 3.12: An example of a correlation graph for a punctured TC. Here $g = 5$, with $np_s = 4$ and $np_p = 5$.

An application example is provided, derived from the NB-TC over GF(64) described in the previous section. It uses the best known convolutional code in terms of distance spectrum, an ARP interleaver, and puncturing masks leading to rate $R = 2/3$. The used ARP interleaver is ARP I from Table 3.2 and the proposed puncturing masks are shown in Table 3.3.

Puncturing mask	Systematic $\mathbf{d}$	Parity $\mathbf{r_1}$	Parity $\mathbf{r_2}$	np_s	np_p
I	11111 11111	00100 01000	01000 10100	4	2
II	11111 11111	01000 01100	00110 00000	4	0

Table 3.3: Puncturing masks optimized for ARP I interleaver from Table 3.2, $K_s = 30$ symbols in GF(64) ($K_b = 180$ bits) and $R = 2/3$. The puncturing period is $M = 10$, np_s and np_p are the minimum number of non-punctured systematic and parity symbols in the minimum correlation cycles.

The two puncturing masks in Table 3.3 differ in the minimum number of parity symbols

in the shortest correlation cycles. In Fig. 3.13, a comparison is conducted between the two resulting NB-TCs, mapped to 4-QAM and 64-QAM constellations. It is shown that puncturing mask I offers better performance than puncturing mask II. The performance gap between the two codes is $\sim$0.8 dB at FER $=10^{-6}$. Better asymptotic performance is obtained with the mask yielding a larger value for np_p. This performance gap validates our proposal related to the selection of the puncturing masks.

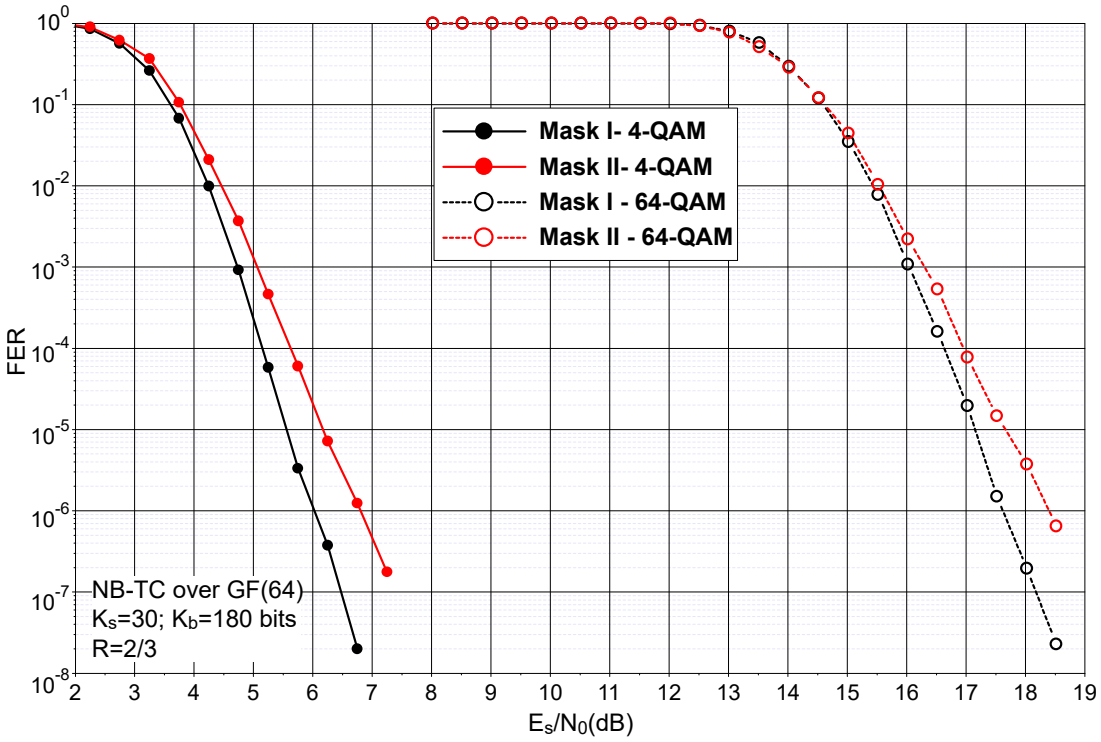

Figure 3.13: Frame error rate performance comparison of two NB-TCs defined over GF(64) with two different puncturing masks for $R=2/3$. Transmission over an AWGN channel with 4-QAM and 64-QAM constellations. $K_s=30$ GF(64) information symbols.

3.2.2 Bit puncturing

When high-order constellations are considered, a new degree of freedom can be investigated. A symbol a in GF(64) is composed of 6 bits, and can be written as $a^5a^4a^3a^2a^1a^0$ with a^5 being the Most Significant Bit (MSB) and a^0 being the Least Significant Bit (LSB). When considering a 64-QAM constellation, bits a^i are protected differently by the constellation. For example, considering the Gray binary mapping shown in Fig. 3.14, three different protection levels can be identified. Bits a^5 and a^4 enjoy the highest protection level, whereas a^1 and a^0 are the bits with the lowest protection level and bits a^3 and a^2 are in between. For illustration purpose, the region with bit a^5 equal to 0 is shaded in red, the region with bit a^2 equal to 0 is shaded in blue and the region with bit a^0 equal to 0 is shaded in green. This non-uniform protection of the bits in a symbol can be exploited to design more elaborate puncturing patterns.

For example, let us consider bit puncturing (BiP) to target coding rate $R=1/2$ for a coded transmission using a NB-TC defined over GF(64) with mother coding rate 1/3 and a 64-QAM constellation.

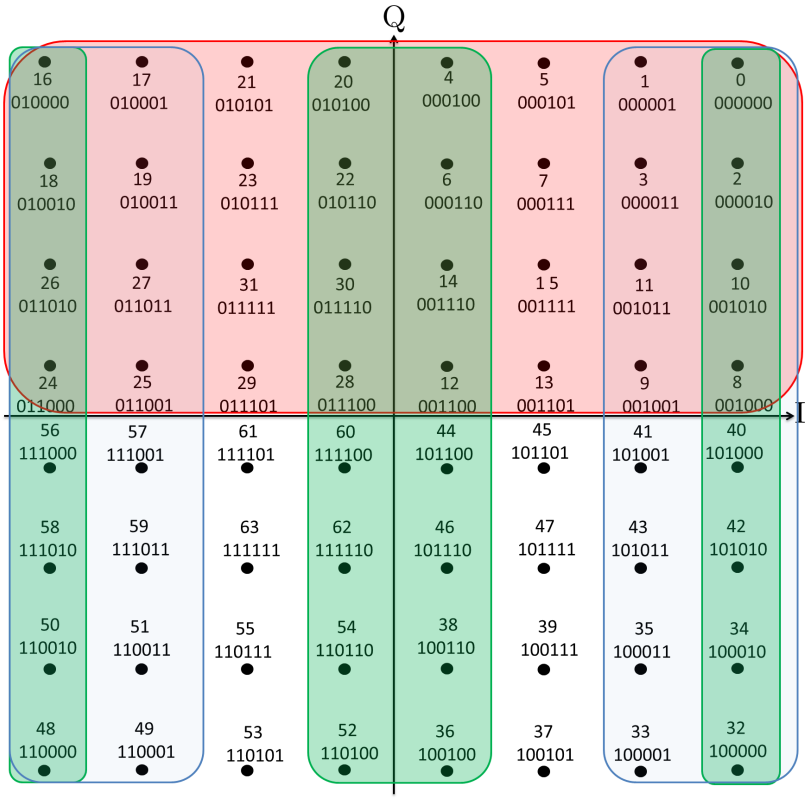

Figure 3.14: 64-QAM constellation with different zones representing the variation of bits a^5, a^2 and a^0.

For TCs, a given coding rate can be obtained with different puncturing ratios of the systematic and parity symbols. Puncturing only parity symbols provides the best performance in terms of convergence threshold while puncturing only systematic symbols is known to lower the error floor. Therefore, the puncturing strategy depends on the target error rate. In our example, we consider puncturing only bits in parity symbols, to keep the SNR in the waterfall region as low as possible.

Fig. 3.15 shows the proposed puncturing pattern at the bit level: no bit of the binary image of the systematic symbols is punctured and systematic bits are transmitted in positions a^5, a^4, which enjoy the highest protection level and in position a^3 which has medium protection. Three bits in each parity symbol are discarded and the others are transmitted in the remaining bit positions – with medium and low protection levels – of the two transmitted symbols. Therefore, each transmitted symbol contains both systematic and parity bits, and guarantees a better protection level for the systematic bits.

The 3 transmitted bits for parity symbols were chosen based on the structure of code C_3, whose parameters are given in Table 2.4. In the trellis of a NB-CC, when DC paths stretch over L trellis sections of a component code, the CSED between two DC sequences X^1 and X^2 is calculated as follows:

$$d^2_{Euc} = \sum_{l=1}^{L} \left(d^2(s_{X^1_l}, s_{X^2_l}) + d^2(p_{X^1_l}, p_{X^2_l}) \right) \tag{3.5}$$

where $s_{X^m_l}$ and $p_{X^m_l}$, $m = 1, 2$, represent the systematic and parity symbol values for sequence X^m at trellis section l.

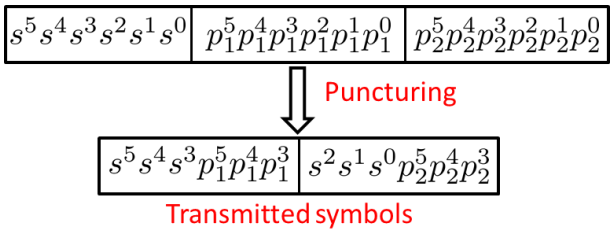

Figure 3.15: Proposed BiP for NB-TCs over GF(64) with mother code rate 1/3 to achieve $R = 1/2$ in a transmission using 64-QAM.

For the component codes designed over different Galois fields as described in Chapter 2, we have observed that the DC paths with the lowest CSED values are obtained for sequences whose competing systematic symbols are mapped to points very close to each other in the constellation[1].

This means that the lowest values of d^2_{Euc} in Eq. (3.5) are obtained for low values of $d^2(s_{X_l^1}, s_{X_l^2})$ and that NB-CCs with good distance spectra are designed in a way to have parity symbols $p_{X_l^1}$ and $p_{X_l^2}$ mapped to points far from each other in the constellation when $s_{X_l^1}$ and $s_{X_l^2}$ are neighbors, as described in Chapter 2. This is the case of code C_3 and Fig. 3.16 illustrates this feature. Fig. 3.16 (a) shows an example of transmitted systematic symbol (15) with its 9 closest neighbors. Fig. 3.16 (b) shows the parity symbols corresponding to the transmitted symbol for code C_3 (symbol 18) and to its neighbors (green circled points). One can observe that these symbols are placed far from each other. For comparison purposes, we have added in Fig. 3.16 (b) the positions of parity symbols corresponding to the transmitted symbol and to its neighbors for code C_1 which has poor distance properties (red circled points). We can observe that the distribution of the parity symbols is not as good as with C_3.

When the code is punctured according to the above-mentioned technique, at the receiver side, 64 Euclidean distances are computed for each received 64-QAM symbol. Then, after a marginalizing and recombining stage, 64 distances can be derived for each systematic symbol, while only 8 distances can be obtained for each punctured parity symbol and, then, only 8 different values are available for terms $d^2(p_{X_l^1}, p_{X_l^2})$ in Eq. (3.5). Therefore, the transmitted parity bits should be selected in such a way that the values of terms $d^2(p_{X_l^1}, p_{X_l^2})$ are as high as possible when $s_{X_l^1}$ and $s_{X_l^2}$ are neighbors in the constellation. In practice, we have adopted a simplified approach: we have chosen the three bits that minimize the number of terms $d^2(p_{X_l^1}, p_{X_l^2})$ equal to zero when the corresponding transmitted systematic symbols are neighbors. It turned out that, for code C_3, this is verified when the 3 MSBs parity bits $p^5_{X_l^m}$, $p^4_{X_l^m}$, $p^3_{X_l^m}$ are transmitted, as shown in Fig. 3.15. Fig. 3.16 (b) shows the 8 regions in the 64-QAM constellation corresponding to the 8 different distance values. Two punctured parity symbols in the same region cannot be distinguished. One can observe that, with these boundaries, not more than two neighbor systematic symbols have their parity in the same region. One can also see that the proposed bit puncturing is much less discriminant for code C_1 since up to 4 parity symbols can end up in the same region.

In order to assess the error correction performance of this BiP strategy, we have simulated rate $R = 1/2$ NB-TCs based on codes in GF(64) C_1 and C_3 and an ARP

[1]This point is developed in Section 4.2 of Chapter 4

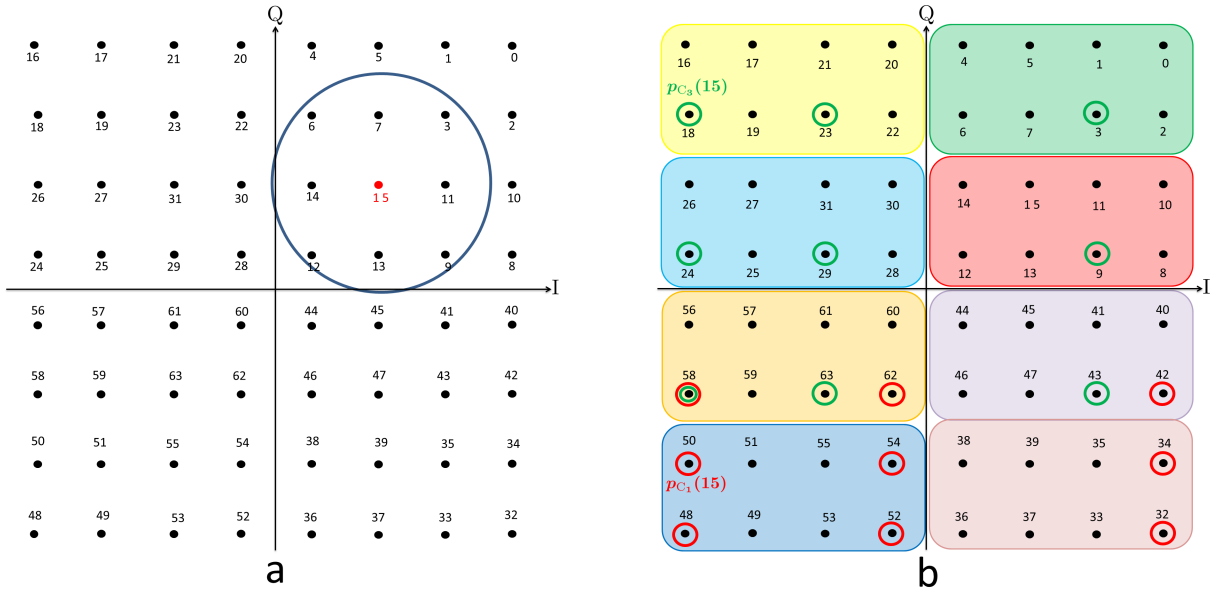

Figure 3.16: (a) Example of a transmitted systematic symbol (15) with its neighborings competing symbols in the 64-QAM constellation. (b) Resulting parity symbols for the systematic competing symbols of (a), for code C_1 (red circles) and code C_3 (green circles).

interleaver using the two proposed puncturing strategies: regular puncturing of one out of every two parity symbols (denoted by SP) and the BiP pattern given in Fig. 3.15. Fig. 3.17 gives the corresponding FER curves for the transmission of blocks of $K_s = 30$ GF(64) symbols in an AWGN channel. It shows that, when the BiP pattern is well designed, it provides a significant coding gain, even in the high and medium FER region (0.5 dB @FER=10^{-1} and 0.75 dB @FER=10^{-7}). As expected, the gain is much less clear for code C_1 due to the reasons mentioned above.

These results show that BiP is promising technique for NB-TCs. However, a more extensive study of this topic has still to be carried out: refinement of the process to select the parity bits to be transmitted, application to a larger range of code rates, preferably compatible with incremental redundancy.

3.3 Distance spectrum evaluation of NB-TCs and computation of the truncated union bounds

The union bounds on the bit and frame error probability [42] are classically used to predict the asymptotic performance of binary turbo codes. The practical computation of the union bound calls for the estimate of the truncated Hamming distance spectrum of the code, since only the first few distance values have a noticeable impact on the obtained results. Impulse-based methods [67, 69] are generally used to provide such an estimate. However, for NB-TCs, the computation of such bounds calls for the estimate of the truncated ED spectrum. Moreover, the ED spectrum largely depends on the particular definition of the component codes of the NB-TC. Therefore, no tool currently exists to predict the asymptotic performance of the newly defined NB-TCs, based on the use of NB-CCs proposed in Chapter 2. In this part of the chapter, we remedy this absence by

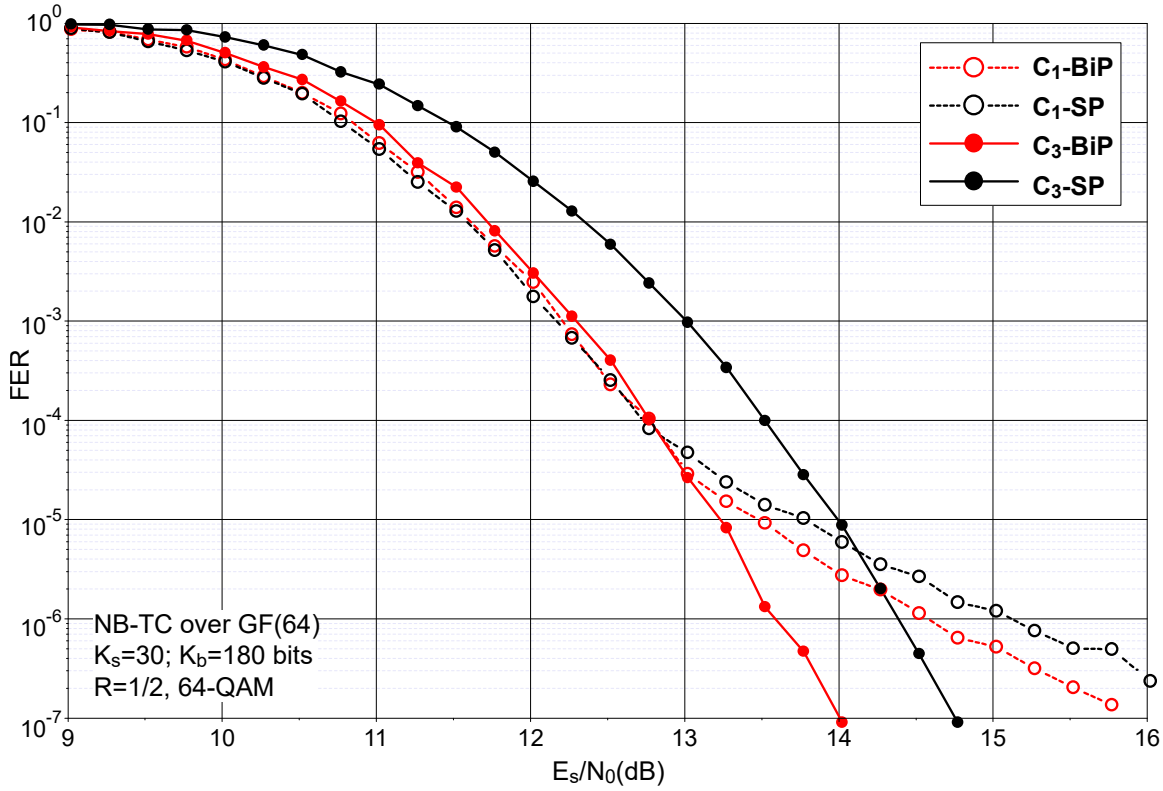

Figure 3.17: Frame error rate performance for NB-TCs over GF(64) with ARP interleaver over an AWGN channel for $K_s = 30$ symbols and $R = 1/2$, mapped to a 64-QAM constellation. Two puncturing strategies are considered: Bit Puncturing (BiP) and Symbol Puncturing (SP).

providing a suitable method to estimate ED spectrum of NB-TCs since the impulse-based methods are not directly applicable for the NB case.

3.3.1 Union bounds formulation for NB codes

The evaluation of the union bound for binary TCs is useful in predicting their behavior at high signal-to-noise ratios [65, 57]. For NB codes, such a union bound can also be derived. Provided that the sequences of data symbols are equiprobable, the frame error probability, P_f, of the code is given by [42]:

$$P_f \leq U_f = \frac{1}{|C|} \sum_{m=1}^{|C|} P_{E_m} \tag{3.6}$$

where C is the set of codewords, $|C|$ represents the cardinality of C ($|C| = q^{K_s}$ for messages of K_s q-ary symbols), and P_{E_m} is the message probability of error when codeword X^m, $m \in \{1, \ldots, |C|\}$ is transmitted.

Since most of the standard high-order modulations (q-QAM, q-APSK,...) do not have the uniform error property [42], the message error probabilities are not ensured to be equal. Therefore, all the P_{E_m} values have to be computed in this case. P_{E_m} is upper bounded by:

$$U_{B_m} = \sum_{k=1,\ k\neq m}^{|C|} P(X^m \rightarrow X^k) \tag{3.7}$$

where $P(X^m \rightarrow X^k)$ is the probability of transmitting codeword X^m and decoding X^k. For a transmission over an AWGN channel, this union bound under maximum likelihood decoding is given by:

$$U_{B_m} = \sum_{k=1,\ k\neq m}^{|C|} Q\left(\frac{d_{m,k}}{2\sigma}\right) \tag{3.8}$$

where $d_{m,k} = \|X^m - X^k\|$ is the Euclidean distance between codewords X^m and X^k, σ is the noise standard deviation and $Q(x)$ is the tail distribution function of the standard normal distribution, defined by $Q(x) = \frac{1}{2}\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)$. The frame error rate is then upper bounded by:

$$\begin{aligned} P_f &\leq \frac{1}{|C|} \sum_{m=1}^{|C|} U_{B_m} \\ &\leq \frac{1}{|C|} \sum_{m=1}^{|C|} \sum_{k=1,\ k\neq m}^{|C|} Q\left(\frac{d_{m,k}}{2\sigma}\right) \end{aligned} \tag{3.9}$$

If each distance $d_{m,k}$ is accounted once, P_f can also be expressed as :

$$P_f \leq \frac{2}{|C|} \sum_{m=1}^{|C|} \sum_{k>m}^{|C|} Q\left(\frac{d_{m,k}}{2\sigma}\right) \tag{3.10}$$

Reordering the distance terms in increasing order: $d_i, i = \{1, \ldots, i_{max}\}$ such that $d_{min} = d_1 < \cdots < d_{i_{max}}$, Eq. (3.10) can also be written as follows:

$$P_f \leq 2 \sum_{i=1}^{i_{max}} r(d_i) Q\left(\frac{d_i}{2\sigma}\right) \tag{3.11}$$

where $r(d_i) = \frac{n(d_i)}{|C|}$, $n(d_i)$ being the multiplicity of ED d_i, that is the number of pairs of codewords with ED d_i.

Note that a similar derivation has already been carried out in [40] for the case of NB parity-check codes.

3.3.2 Identification of low-distance DC sequences for a NB-TC

The computation of the complete union bound requires the enumeration of the complete ED spectrum of the code, which is not practically feasible for large values of q and/or K_s. Therefore, in practice, for a given family of codes, one has to identify the type of codewords chosen by the decoder when it is in error, in order to limit the search process.

The adopted structure of NB-CCs has a fully connected trellis that reduces the evaluation process to a limited number of trellis stages, at most 3 in practice, as mentioned in Section 2.4.2. As a consequence, in most cases, the turbo decoder is in error when errors introduced by the channel create short – i.e. length-2 or length-3 – erroneous sequences in the trellises of both component decoders. To enumerate such error events, we first have to identify short DC sequences in the first component code and their corresponding short DC sequences after interleaving, which we refer to as turbo-DC sequences. No method currently exists to achieve an accurate enumeration for the NB-TC as a whole, due to the effect of the interleaver. To this end, we identified the shape of such error events and the underlying reasons that lead to their occurrence. We found out two main sources: correlation cycles and spread-resilient error sequences.

3.3.2.1 Effect of short correlation cycles on the error events of NB-TCs

In [10], correlation graphs were introduced in the context of binary TCs, and a new interleaving design was proposed, targeting the maximization of the length of the shortest cycles, denoted by the girth g. In Section 3.1.4, different cycle types and constraints were identified, where their impact on correlation, and hence performance, was evaluated and reported.

In the case of NB-TCs, due to the particular nature of the component codes with fully connected trellises, cycles due to the interleaver design seem to play a central role in the occurrence of short turbo-DC sequences, largely impacting performance. Without loss of generality and only for illustration purposes, Fig. 3.18 provides an example of a cycle and DC sequences between the two trellises of the components codes of a NB-TC, where the all-zero sequence X^1 ($\tilde{X}^1$ in the interleaved order) is assumed to be transmitted, and sequences I to IV represent the DC sequences created by the cycle. The limitation to the all-zero sequence is only taken for illustrative purposes, since all other sequences and initial states are to be considered for the estimation of the spectrum. X^2 ($\tilde{X}^2$ in the interleaved order) denotes the competing DC sequence. The interleaver defines the connection linking the trellis of the first component code to the trellis of the second component code. Four interleaving connections are shown in Fig. 3.18. In this particular

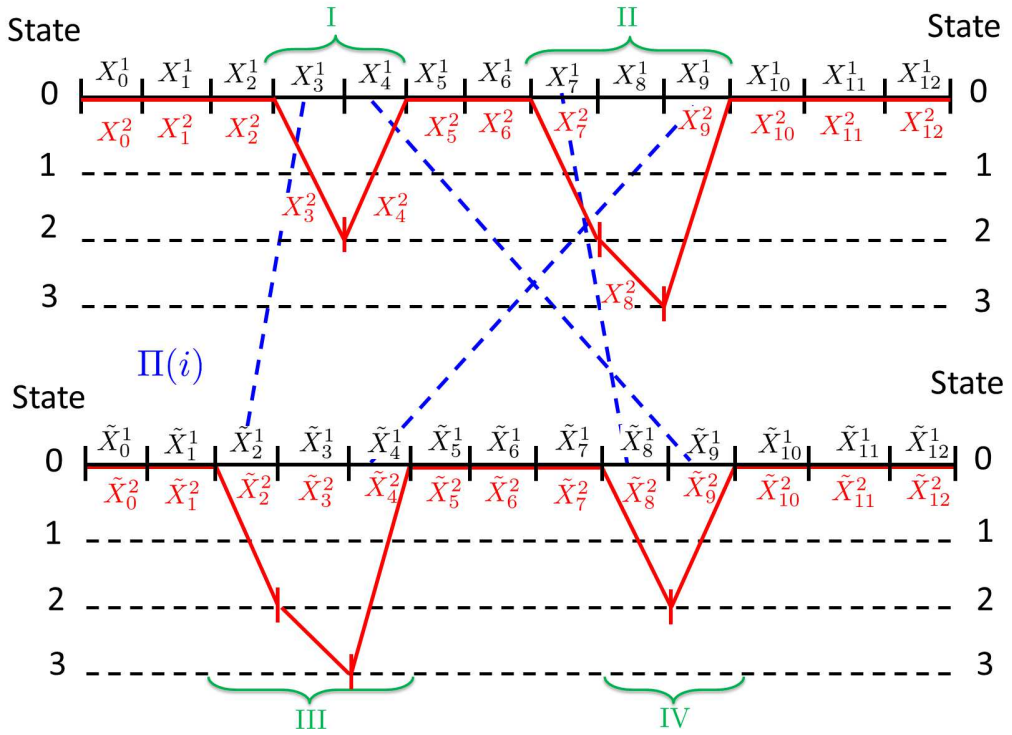

Figure 3.18: Example of type-A turbo-DC sequences and correlation cycle for a $q=4$ NB-TC. The all-zero sequence is assumed to be transmitted. The length of the correlation cycle is $g=6$.

example, although neighboring symbols in the natural order are spread out by the interleaver, the applied permutation simultaneously creates a short correlation cycle of length 6 and a pair of turbo-DC sequences, comprising the all-zero sequence and its competing sequence consisting of (X_3^2, X_4^2) and (X_7^2, X_8^2, X_9^2) in the natural order and $(\tilde{X}_2^2, \tilde{X}_3^2, \tilde{X}_4^2)$ and $(\tilde{X}_8^2, \tilde{X}_9^2)$ in the interleaved order. Due to the fully connected recursive nature of the component NB-CC, such short correlation cycles can be frequently encountered if the interleaver is not specifically designed to avoid them. More precisely, we can always find a sequence X^2 (respectively $\tilde{X}^2$), competing with X^1 (respectively $\tilde{X}^1$) such that X^1 and X^2 (respectively $\tilde{X}^1$ and $\tilde{X}^2$) share the same systematic symbols in all frame positions except in the positions belonging to a short correlation cycle: in the example of Fig. 3.18, these positions correspond to indexes $\{3,4,7,8,9\}$ in the natural order and $\{2,3,4,8,9\}$ in the interleaved order. When such a scenario occurs, since sequences X^1 and X^2 (respectively, their interleaved versions $\tilde{X}^1$ and $\tilde{X}^2$) only differ in a low number of trellis sections, they are highly likely to show a low ED. This type of turbo-DC sequences caused by short correlation cycles is denoted by type-A sequences in the remainder of this work.

3.3.2.2 Error events resilient to interleaver spread

Another type of turbo-DC sequences, denoted as type-B sequences, is depicted in Fig. 3.19. They consist of a pair of DC sequences of length-(l_1+2) in the natural order and a pair of DC sequences of length-(l_2+2) in the interleaved order. Due to the structure of the component codes, it is ensured that X^1 and X^2 (respectively $\tilde{X}^1$ and $\tilde{X}^2$) differ by at least the systematic symbols when they diverge and converge: in Fig. 3.19, they differ

at positions $\{3,5\}$ in the natural order, and at positions $\{2,10\}$ in the interleaved order. However, there exist some sequences X^1 and X^2 (respectively $\tilde{X}^1$ and $\tilde{X}^2$) with the same systematic symbols in the l_1 (respectively l_2) remaining positions. In this case, these positions only contribute to the ED through their parity symbols. For low values of $L = l_1 + l_2$, low ED values are likely to be achieved.

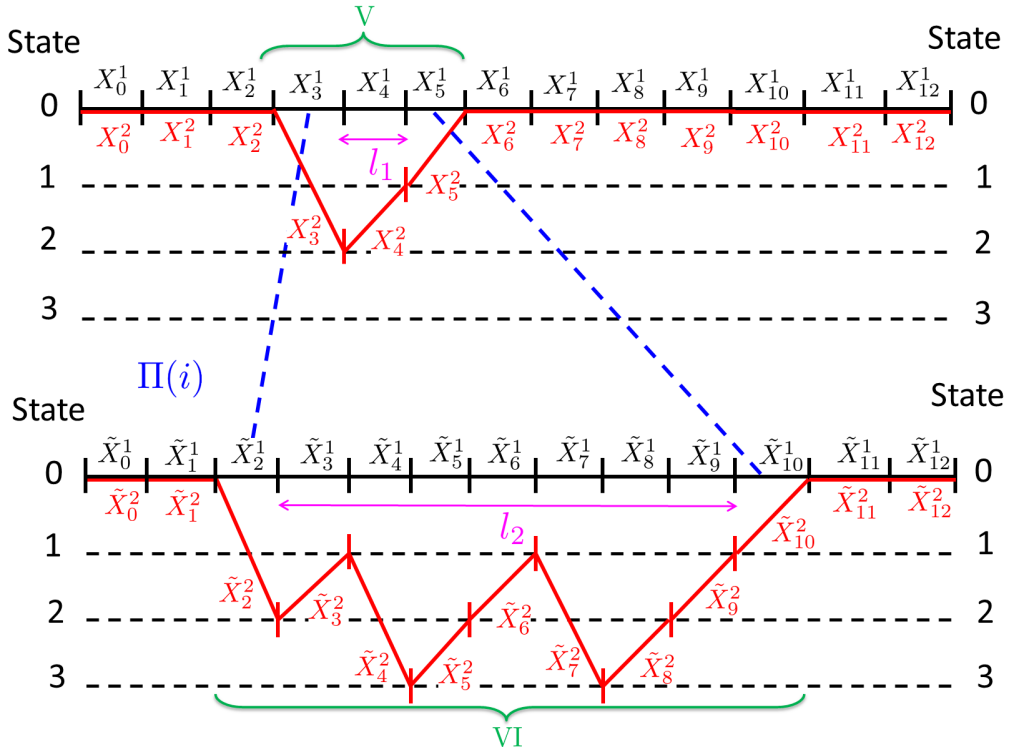

Figure 3.19: Example of type-B turbo-DC sequences for a $q = 4$ NB-TC. A length-$(l_1 + 2)$ DC sequence is shown in the natural order and a length-$(l_2 + 2)$ DC sequence is shown in the interleaved order.

3.3.3 Estimation of the truncated distance spectrum of NB-TCs

The proposed method first identifies all the type-A sequences corresponding to the shortest correlation cycles created by the interleaver. Then, depending on the minimum ED obtained at the first step, we set an upper limit on the value of L in the identification step of type-B DC sequences related to the interleaver spread. Finally, after sorting, the truncated distance spectrum of the turbo code is obtained.

3.3.3.1 Computation of the Euclidean distance of type-A sequences

For type-A sequences, the first task is to determine the lowest value of the correlation girth, in order to calculate the corresponding minimum achievable ED and then estimate the corresponding multiplicity.

The computation of the minimum achievable ED for a given correlation girth value involves finding the different possible turbo-DC sequence shapes for the considered correlation cycles and dividing them into elementary DC sequences, which can be classified into two groups: length-2 DC sequences and DC sequences of length $l > 2$. Then, the

minimum achievable ED can be computed for each turbo-DC sequence shape by adding up the EDs of the elementary DC sequences.

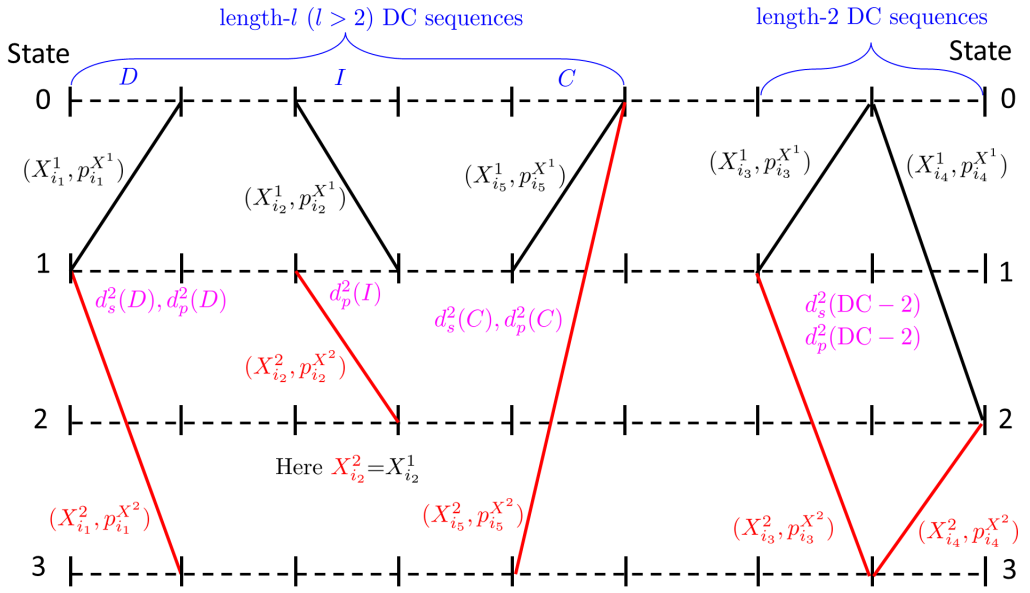

Figure 3.20: Example of length-2 DC sequence and the decomposition of length-l, $l > 2$ DC sequences used for the computation of the minimum achievable ED for a given turbo-DC sequence shape.

For DC sequences longer than 2 (II and III in Fig. 3.18), the minimum ED values can be obtained by splitting the sequences into three types of trellis sections: a diverging section $\boldsymbol{D}$, a converging section $\boldsymbol{C}$ and a number of intermediate sections $\boldsymbol{I}$, depending on the sequence length. The three types of trellis sections are illustrated in Fig. 3.20 for a NB-CC defined over GF(4). For a given NB-CC, the minimum value for the squared Euclidean distance (SED) related to each type of trellis section, $d^2(\boldsymbol{D})$, $d^2(\boldsymbol{C})$ and $d^2(\boldsymbol{I})$ are known and:

$$d^2(\mathrm{DC}-l) = d^2(\boldsymbol{D}) + (l-2)d^2(\boldsymbol{I}) + d^2(\boldsymbol{C}) \tag{3.12}$$

In the case of a rate-1/2 NB-CC code, each term in Eq. 3.12 is the sum of two distance terms related to the systematic and parity symbols s and p: $d^2(.) = d^2_s(.) + d^2_p(.)$.

Actually, due to the design rules of the NB-CC code, and especially, due to the fully-connected property of the trellis, the distance term $d^2(\boldsymbol{I})$ is at its minimum when all the systematic symbols are identical for the intermediate sections of the competing sequences. Therefore $d^2_s(\boldsymbol{I})$ is equal to 0 and only the parity symbols have to be taken into account for these sections. Therefore, the minimum achievable SED for length-l ($l > 2$) DC sequences is equal to:

$$d^2(\mathrm{DC}-l) = \left(d^2_s(\boldsymbol{D}) + d^2_p(\boldsymbol{D})\right) + (l-2)d^2_p(\boldsymbol{I}) + \left(d^2_s(\boldsymbol{C}) + d^2_p(\boldsymbol{C})\right) \tag{3.13}$$

which can also be written as follows when separating systematic and parity terms:

$$\begin{aligned} d^2(\mathrm{DC}-l) = & \underbrace{\left(d^2_s(\boldsymbol{D}) + d^2_s(\boldsymbol{C})\right)}_{d^2_s(\mathrm{DC}-l)\text{ systematic term}} \\ & + \underbrace{\left(d^2_p(\boldsymbol{D}) + (l-2)d^2_p(\boldsymbol{I}) + d^2_p(\boldsymbol{C})\right)}_{d^2_p(\mathrm{DC}-l)\text{ parity term}} \end{aligned} \tag{3.14}$$

For length-2 DC sequences such as I and IV in Fig. 3.18, this decomposition is not valid since the NB-CCs are designed to guarantee that the minimum ED of a length-2 sequence is greater than the sum of the minimum squared Euclidean distances of a diverging section $\boldsymbol{D}$ and a converging section $\boldsymbol{C}$ taken separately. Therefore, length-2 DC sequences have to be handled apart and $d^2(\mathrm{DC}-2)$ is known for a given NB-CC. It can also be written as:

$$d^2(\mathrm{DC}-2) = d_s^2(\mathrm{DC}-2) + d_p^2(\mathrm{DC}-2) \tag{3.15}$$

Let us illustrate the proposed approach with the example in Fig. 3.18. The girth-6 cycle is caused by pairs of turbo-DC sequences composed of length-2 and length-3 DC sequences, for both component codes, in the natural and interleaved orders. The minimum achievable SED $d_{g=6}^2$ for this shape of turbo-DC sequences is obtained by adding up the minimum achievable SEDs of the constituent DC sequences I, II, III and IV. However, since the systematic symbols are only transmitted once for both component codes, they should not be counted twice. Therefore, only the parity terms are considered for sequences I and II or for sequences III and IV. Then:

$$\begin{aligned} d_{g=6}^2 = &\underbrace{d^2(\mathrm{DC}-2)}_{I} + \underbrace{d^2(\mathrm{DC}-3)}_{II} \\ &+ \underbrace{d_p^2(\mathrm{DC}-3)}_{III} + \underbrace{d_p^2(\mathrm{DC}-2)}_{IV} \end{aligned} \tag{3.16}$$

In practice, once the minimum correlation girth value g_{min} has been determined, all the type-A turbo-DC sequences with a correlation girth equal to g_{min} are enumerated and their corresponding SED are computed using Eq. (3.14) and/or Eq. (3.15).

The number and/or the length of elementary DC sequences in type-A turbo-DC sequences increases with the girth value. Therefore, the minimum achievable SED obtained from type-A sequences is an increasing function of the girth. In order to get a more accurate value of the bound, the proposed approach can be extended to higher correlation girth values, but the search complexity is then significantly increased. The choice of the maximum girth value to be considered in the search algorithm sets a compromise between the tightness of the bound and the sequence enumeration complexity.

3.3.3.2 Computation of the minimum achievable Euclidean distance of type-B sequences

Type-B turbo-DC sequences are composed of two pairs of DC sequences of length l_1+2 and l_2+2. For this type of sequences, provided that the interleaver is well designed, the low EDs are obtained when all the systematic symbols in the competing sequences are identical, except in the diverging and converging trellis sections. Therefore, using the same notations as in the previous section and counting the systematic terms only once, according to Eq. (3.14), the minimum achievable SED for the type-B turbo-DC sequences of Fig. 3.19 can be expressed as:

$$\begin{aligned} d_{l_1+l_2}^2 = &\underbrace{\left(d_s^2(D) + d_s^2(C)\right)}_{V \text{ or } VI} + \underbrace{\left(d_p^2(D) + l_1 d_p^2(I) + d_p^2(C)\right)}_{V} \\ &+ \underbrace{\left(d_p^2(D) + l_2 d_p^2(I) + d_p^2(C)\right)}_{VI} \end{aligned} \tag{3.17}$$

All the distance terms in this expression only depend on the NB-CC parameters and are known. In order to limit the complexity of the enumeration process for type-B turbo-DC sequences, an upper limit has to be set on the value of $L = l_1 + l_2$ to be considered. If we want all pairs of type-B sequences with minimal SED d^2_{min} to be considered in the enumeration, the value of L should then be chosen so that $d^2_{min} \geq d^2_{l_1+l_2}$. Rearranging Eq. (3.17), we obtain:

$$L \leq \frac{d^2_{min} - \left(d^2_s(D) + d^2_s(C)\right) - 2\left(d^2_p(D) + d^2_p(C)\right)}{d^2_p(I)} \tag{3.18}$$

In practice, the value of d^2_{min} used to determine L_{max} is the minimum achievable SED value found when enumerating type-A sequences, which is guaranteed to be greater than or equal to the minimum SED of the NB-TC. Type-B sequences with length up to L_{max} are then enumerated and their corresponding SED are computed using Eq. (3.17).

In order to get a more accurate value of the bound, the enumeration can be extended to longer type-B sequences, but the search complexity is then significantly increased. The choice of the L_{max} value to be considered in the search algorithm sets a compromise between the tightness of the bound and the sequence enumeration complexity.

3.3.3.3 Computation of the truncated union bound

From the enumeration of type-A and type-B turbo-DC sequences, a truncated union bound can be computed according to Eq. (3.11).

In practice, for each considered subset S of type-A and type-B turbo-DC sequences, their enumeration provides the corresponding ED sub-spectrum. Successive distance values $d_i, i \geq 1$ with their multiplicities, $n_S(d_i)$, are provided for each subset S.

However, the computation of the term $r(d_i) = \frac{n(d_i)}{|C|}$ in Eq. (3.11) cannot be directly derived from these truncated sub-spectra, because all the enumerated types of turbo-DC sequences do not have the same weight in the bound formula. In fact, each considered pair of turbo-DC sequences represents a larger set of codewords. In the example of Fig. 3.18, the $K_s - 5$ systematic symbols labeling the trellis sections placed before, after and in-between the DC sections should be equal in both competing paths but could take any value in GF(q). Therefore, the pair of type-A turbo-DC sequences shown in Fig. 3.18 accounts for q^{K_s-5} different pairs of codewords, since the total length of the DC sections is 5, for both constituent codes.

Similarly, for type-B sequences, each pair of turbo-DC sequences accounts for $q^{K_s-(2+l_1+l_2)}$ pairs of codewords, provided that the inner sections of the elementary DC sequences in the interleaved and non-interleaved order do not have any systematic symbols in common. If these sequences have m common systematic symbols, only $K_s - (2 + l_1 + l_2 - m)$ systematic symbols can be freely chosen instead of $K_s - (2 + l_1 + l_2)$ and each pair of turbo-DC sequences only accounts for $q^{K_s-(2+l_1+l_2-m)}$ pairs of codewords.

Therefore, in practice, the multiplicity $n(d_i)$ related to distance d_i is replaced in Eq. (3.11) by :

$$n(d_i) = \sum_S n_S(d_i) q^{K_s - l_S} \tag{3.19}$$

where S spans all the subsets of type-A and type-B sequences considered in the enumeration and $q^{K_s-l_S}$ is the multiplicative factor related to each subset, as explained above.

Noting that $|C| = q^{K_s}$, which represents the total number of codewords, the normalized multiplicity is in practice computed as:

$$r(d_i) \approx \frac{\sum_S n_S(d_i) q^{K_s - l_S}}{q^{K_s}} \tag{3.20}$$

Please note that, depending on the cycle length considered for enumerating type-A sequences and the L_{max} value considered for the enumeration of type-B sequences, the multiplicities of larger ED may be underestimated. Despite this simplification and since the Q function decreases rapidly with increasing distance values, limiting the study to the lowest EDs is often sufficient to predict the asymptotic performance of the considered codes.

3.3.4 Numerical results

Following the method proposed in Section 3.3.2, the truncated distance spectra of two NB-TCs over GF(64), mapped to 64-QAM and 4-QAM constellations have been estimated. Two memory-1 NB-CCs are considered (C_1 and C_3 from Table 2.4) and are associated with 64-QAM and 4-QAM constellations following the mapping rule given in Table 2.3. The code rate is $R = 1/3$ for all the simulated TCs. Information frame size is $K_s = 30$ GF(64) symbols (equivalent to 180 bits) when the codewords are mapped to a 4-QAM constellation and $K_s = 160$ GF(64) symbols (equivalent to 960 bits) in the case where they are mapped to a 64-QAM constellation. We used an ARP interleaver whose parameters are given in Table 3.4.

ARP	S_{min}	Girth	P	Q	$(S(0),...,S(Q-1))$
I	6	4	19	3	(0,25,14)
II	8	8	143	8	(0,150,62,10,110,22,20,98)

Table 3.4: ARP interleaver parameters for NB-TCs over GF(64) with $R = 1/3$, $K_s = 30$ symbols for ARP-I and $K_s = 160$ symbols for ARP-II (ARP equation: $\Pi(i) = (Pi + S(i \bmod Q)) \bmod K_s$).

For codes mapped to a 4-QAM constellation, the minimum girth value used to enumerate sequences of type-A is $g = 4$. For type-B sequences, the obtained values of L_{max} depend on the constituent encoders: $L_{max} = 10$ for C_1, and $L_{max} = 11$ for C_3. When a 64-QAM constellation is considered, the interleaver achieves in a minimum girth $g = 8$ as shown in Table 3.4. The values of L_{max} used to enumerate type-B sequences are: $L_{max} = 9$ for C_1, and $L_{max} = 32$ for C_3. Simulations were performed for the codes in Table 3.5 over an AWGN channel. Truncated union bounds were calculated following Eq. (3.11) with the truncated distance spectra shown in Table 3.5.

Figures 3.21 and 3.22 show the corresponding error correction performance in terms of FER as a function of E_s/N_0. Truncated union bounds are shown to approach the NB-TCs simulated performance in the error floor region. These results validate the proposed method for the calculation of the NB-TCs truncated distance spectra and union bounds. Moreover, this method can be adopted as a design tool for the identification of the best NB-TCs schemes for low error rate scenarios.

Constellation	4-QAM		64-QAM	
Code	C_1	C_3	C_1	C_3
(a_1, a_2, a_3)	(41,2,0)	(31,5,18)	(41,2,0)	(31,5,18)
d_1^2	22	34	1.04	2.57
$r(d_1)$	6.1e-03	4.5e-04	8.5e-04	1.4e-08
d_2^2	24	38	1.23	2.76
$r(d_2)$	0.25	5.9e-07	7.3e-06	1.4e-08
d_3^2	26	40	1.33	2.95
$r(d_3)$	9.5e-06	3.8e-06	1.4e-08	5.8e-05

Table 3.5: Truncated distance spectra for $R = 1/3$ NB-TCs over GF(64) mapped to a 4-QAM ($K_s = 30$ symbols) and 64-QAM ($K_s = 160$ symbols) constellations. $d_i^2, i = 1,2,3$ are the 3 first SED terms of the NB-TC spectrum, and $r(d_i)$ represents the normalized multiplicity of d_i.

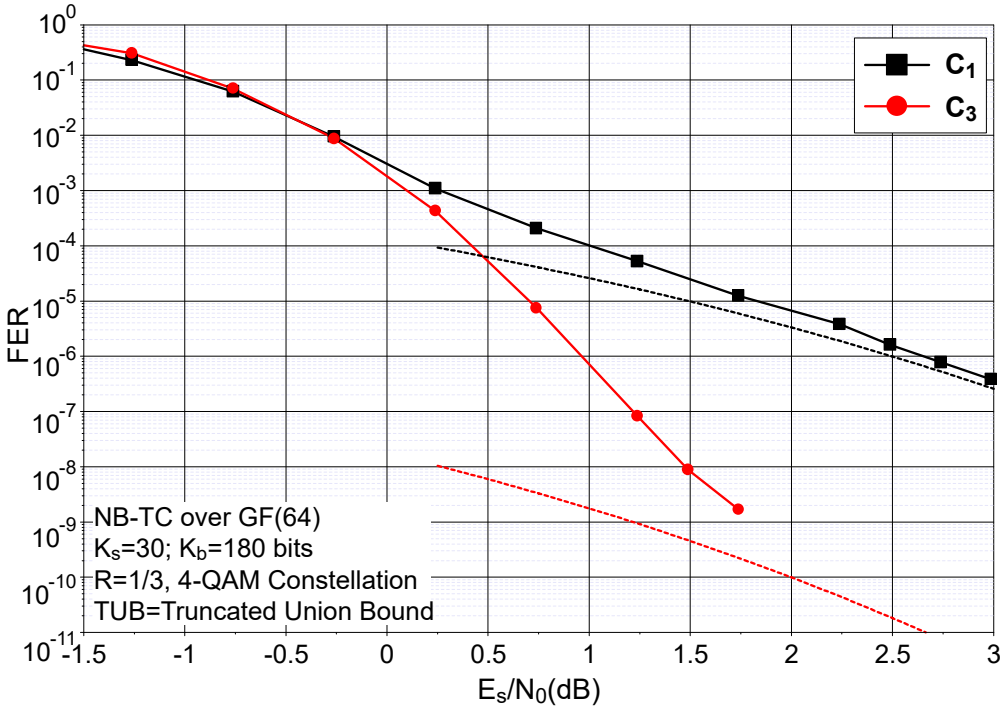

Figure 3.21: FER performance evaluation of the NB-TCs of Table 3.5, defined over GF(64) and mapped to a 4-QAM constellation, with their corresponding TUBs. ARP-I interleaver from Table 3.4 is used. $R = 1/3$ and $K_s = 30$ symbols. AWGN channel, 8 decoding iterations with the scaled Max-Log-MAP algorithm.

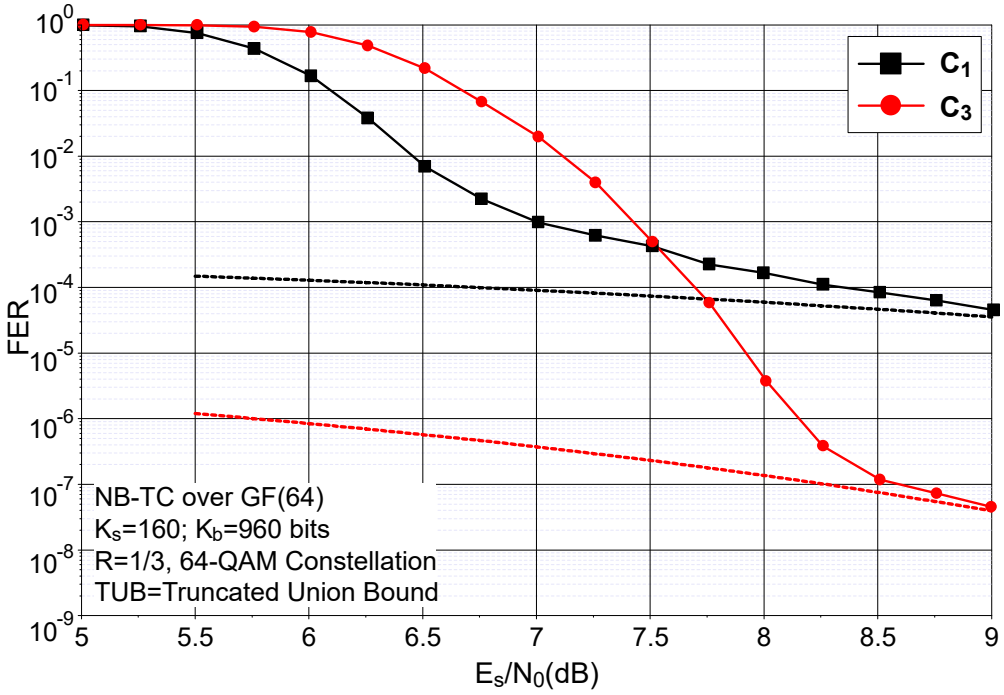

Figure 3.22: FER performance evaluation of the NB-TCs of Table 3.5, defined over GF(64) and mapped to a 64-QAM constellation, with their corresponding TUBs. ARP-II interleaver from Table 3.4 is used. $R = 1/3$ and $K_s = 160$ symbols. AWGN channel, 8 decoding iterations with the scaled Max-Log-MAP algorithm.

3.4 Conclusion

In this chapter, the general structure of NB-TCs is investigated. The effect of different interleaving criteria on the NB-TC performance are studied by applying well-known permutation families. We observed that NB-TCs interleavers are governed by the same principles as their binary counterparts and ARP interleavers are proved to provide high spread and girth values as well as high SEDs. Furthermore, a method is proposed to further mitigate the effect of low correlation cycles, by reducing the impact of information exchange in the decoding process of such cycles. Also, puncturing is investigated in this chapter, and two different methods are proposed: puncturing GF(q) symbols or bit puncturing. Finally, a method to compute the truncated union bound of NB-TCs is proposed.

The study of the correlation mitigation in the decoding process was carried out jointly with Ronald Garzón Bohórquez and was published at the 10$^{\text{th}}$ International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018):

R. Garzón Bohórquez, R. Klaimi, C. Abdel Nour and C. Douillard, "Mitigating Correlation Problems in Turbo Decoders", in 10$^{\text{th}}$ International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018), Hong Kong, Dec. 2018.

Also the proposed method to estimate the distance spectra and the definition of the union bounds in the case of NB-TCs was submitted for publication to the IEEE Transactions on Communications:

R. Klaimi, C. Abdel Nour, C. Douillard and J.Farah "Union bound evaluation for Non-Binary Turbo Codes", submitted to IEEE Trans. on Commun.

Chapter 4

Enhancing performance of non-binary turbo codes via symbol transformation

Amongst the different techniques considered in the literature to improve the error correction performance of existing FEC codes, precoding has already been investigated in the context of TCs. For instance, in [38], precoding was used to lower the iterative decoding threshold of regular LDPC codes and precoding gains up to almost 0.3 dB were observed. On another note, it was shown in [90] that accumulate-repeat-accumulate (ARA) codes can be seen as precoded repeat-accumulate (RA) codes: precoding a portion of the input bits with a rate-1 accumulator can significantly lower the iterative decoding threshold of an RA code. It turns out that TCs are closely related to RA codes since a classical TC can be seen as a serial concatenation of a repetition 2 code and of an RSC code separated by an interleaver. Therefore, in [91], a precoding scheme inspired by the ARA codes was used to precode a TC and precoding gains in the order of 0.4 dB were demonstrated. Later, the authors of [92] investigated alternative precoding structures for TCs that can provide several trade-offs between the convergence threshold and the error floor of a TC.

As far as NB-TCs are concerned, we did not consider adding a precoder to the structure studied in the previous chapter, since this would add non-negligible complexity at the decoder side. However, we searched an alternative means of pre-processing the input data in order to improve the error correction performance of the code, without noticeably increasing the decoding complexity. Actually, the cardinality of the used Galois field offers a new degree of freedom compared with binary codes. Inspired by the intra-symbol permutation introduced for double-binary TCs [74], we found a means to modify the input data of one of the component encoders in order to lower the error floor of NB-TCs without entailing any increase of their convergence threshold. The resulting codes were simulated and compared with the state-of-the-art binary and non-binary codes.

This chapter is organized as follows. Section 4.1 describes the modified structure of NB-TCs studied in this chapter. It is based on the insertion of a symbol transformation before one of the component NB RSC encoders. Section 4.2 is devoted to the design criterion for this transformation. The practical application of this criterion in the case of a code defined over GF(64) and mapped to a 64-QAM constellation is presented in Section 4.3 and in Section 4.4 when the same code is mapped to a 4-QAM constellation. The corresponding results are described in Section 4.5 and Section 4.6 shows the error

correcting performance of the obtained NB-TCs, compared to state-of-the-art FEC codes. Finally, Section 4.7 concludes the chapter.

4.1 Modified structure of the NB turbo encoder

Fig. 4.1 shows the structure of the modified NB-TC considered in this chapter. A symbol transformation block is added to the initial conventional structure in order to change the values of the NB symbols encoded by the second (Fig. 4.1 (a)) or first (Fig. 4.1 (b)) component RSC encoder.

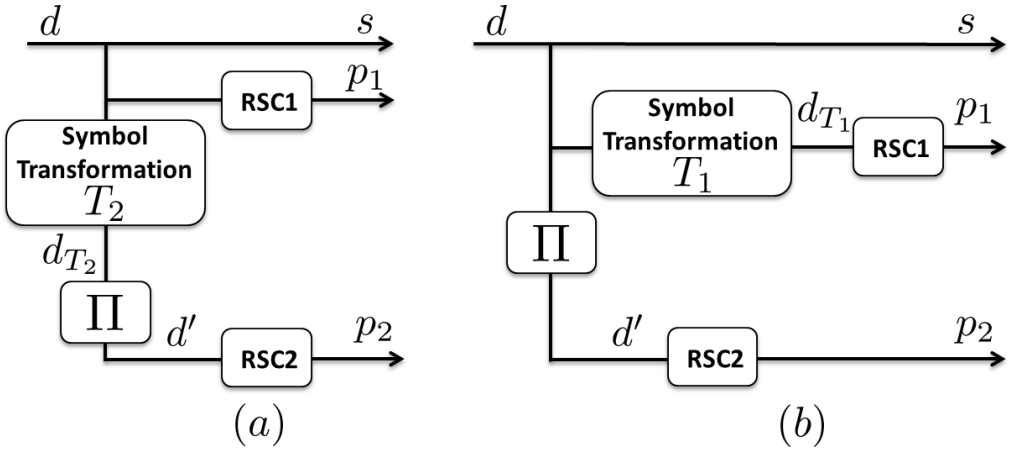

Figure 4.1: Two possible revisited structures of NB-TCs, including the symbol transformation block in different positions.

For a NB-TC defined over GF(q), the transformation block defines a bijection T: GF(q) $\rightarrow$ GF(q). In the case of Fig. 4.1 (a), all the elements of the input sequence d are transformed by T_2, resulting in d_{T_2}, which is then permuted by the interleaver before being encoded by the second encoder RSC2. In this scheme, the symbol transformation block could have also been placed after the interleaver, without any impact on the sequence d' at the input of RSC2. In Fig. 4.1 (b), the symbol transformation is applied to the input sequence encoded by the first RSC encoder. Any of these two schemes can be studied, since the scheme in Fig. 4.1 (a) can be obtained from the one in Fig. 4.1 (b) by applying inverse transformation T_1^{-1} to the symbols of sequence d before being encoded by both component RSC encoders and, conversely, the scheme of Fig. 4.1 (b) can be obtained from the one of Fig. 4.1 (a) by applying inverse transformation T_2^{-1} to the symbols of sequence d before being encoded by both component RSC encoders. In the remainder of this work, the structure of Fig. 4.1 (a) is chosen as the study template and the transformation is simply denoted by T.

The proposed transformation complements the interleaver since the interleaver does not change the values of the symbols. It only re-orders them, whereas the transformation actually changes the set of the symbol values in the data sequence.

In practice, the transformation can be implemented at the encoder side using a lookup table that connects all the symbols of GF(q) in a one-to-one correspondence. At the decoder side, no specific decoding needs to be added: look-up tables implementing function T or T^{-1} have to be inserted at the appropriate places to ensure that the systematic, parity and extrinsic symbols at the input of each SISO decoder all refer to the same transformed or non-transformed data.

4.2 Symbol transformation design criterion

As shown earlier in Chapter 2, due to the accumulator structure of the NB-CCs used as component codes, their trellises are fully connected and DC sequences as short as two trellis stages cannot be avoided.

Short DC sequences with low CSEDs – we also call such sequences "competing sequences" in what follows – often cause some errors in a component decoder but the iterative decoding process helps to correct most of these errors using the extrinsic information related to the involved systematic symbols coming from the other component decoder. However, the turbo decoder may not be able to correct these errors when a transmitted symbol contributes to short DC sequences with low CSEDs in both component trellises with the same competing symbols in both pairs of DC sequences.

Let us illustrate this issue in the case of length-2 DC sequences: let us consider X^1 and X^2 a pair of length-2 DC sequences for the first component code (non-interleaved order) and Y^1 and Y^2 a pair of length-2 DC sequences for the second component code (interleaved order). X^m and Y^m, $m=1,2$, consist of two consecutive couples of systematic and parity symbols, as shown in Fig. 4.2: $s_{X_l^m}/s_{Y_l^m}$ and $p_{X_l^m}/p_{Y_l^m}$, $m=1,2$, $l=1,2$, represent the systematic and parity symbol values respectively for sequence X^m/Y^m at trellis section l.

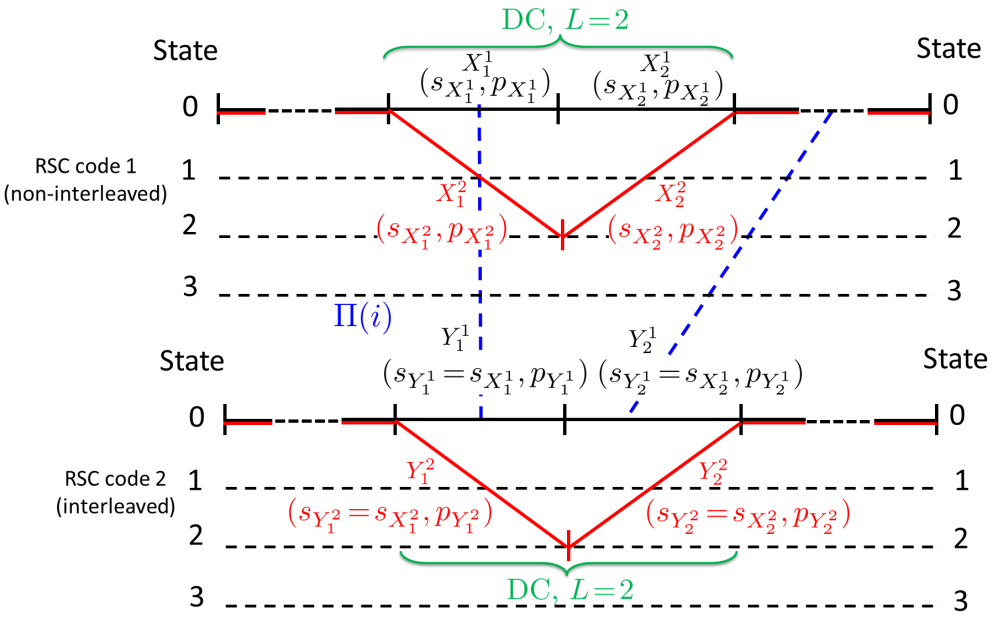

Figure 4.2: Example of competing DC sequences of length $L=2$ in the trellises of both RSC codes of a NB-TC over GF(4).

As explained in Chapter 3, the NB-TC interleaver is designed to guarantee high spread values. Therefore, if systematic symbol $s_{X_1^1}$ is interleaved to $s_{Y_1^1}$, $s_{X_2^1}$ should not be interleaved to $s_{Y_2^1}$ due to the interleaver spread. However, it may happen that the value of systematic symbol $s_{Y_2^1}$ is equal to $s_{X_2^1}$ even if $s_{X_2^1}$ is not interleaved to $s_{Y_2^1}$. In this case, the same pattern of short DC sequences with a low CSED can be simultaneously observed in the trellises of both component codes: $Y^2=X^2$. Then, if the CSED between X^1 and X^2 is low and if X^1 is the transmitted sequence, there is a high probability that the decoder

selects sequences $X^2 = Y^2$ instead of $X^1 = Y^1$, thus causing decoding errors. The same effect can also be observed with longer DC sequences, as far as their CSED remains low.

In the general case, when DC paths stretch over L trellis sections of a component code, the CSED between two DC sequences X^1 and X^2 is calculated as follows:

$$d^2_{Euc1} = \sum_{l=1}^{L} \left(d^2(s_{X_l^1}, s_{X_l^2}) + d^2(p_{X_l^1}, p_{X_l^2}) \right) \tag{4.1}$$

When a transformation function is introduced, the DC sequences at the second component code shown in Fig. 4.2 are changed accordingly, as illustrated in Fig. 4.3.

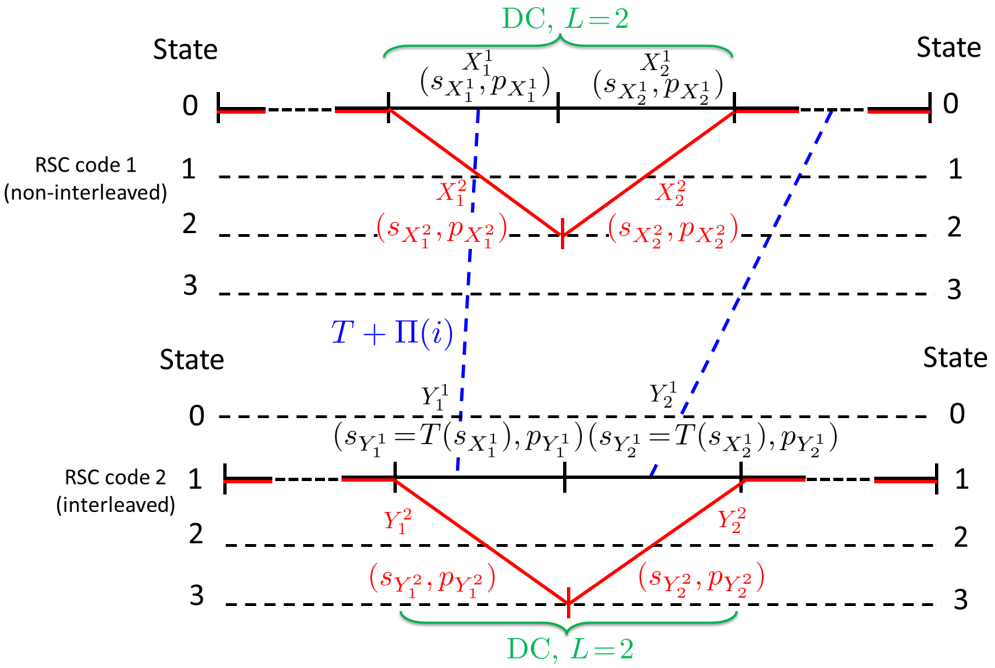

Figure 4.3: Example of competing DC sequences of length $L = 2$ in the trellises of both RSC codes of a NB-TC over GF(4) applying a transformation T.

For the component codes designed over different Galois fields as described in Chapter 2, we have observed that the DC paths with the lowest CSED values are obtained for sequences whose competing systematic symbols (i.e. $s_{X_l^1}$ and $s_{X_l^2}$) are mapped to points very close to each other in the constellation. For instance, in the case of NB-CC codes C_1, C_2 and C_3 previously defined over GF(64) and mapped to a 64-QAM constellation (see Table 2.4), we have enumerated the competing DC sequences having distances $d \leq d_2$, where d_2 is the second lowest distance of the distance spectrum of the considered code. We have observed that the competing systematic symbols were always mapped to the constellation points whose Euclidean distance was lower than or equal to $D_2 = \sqrt{2}D_1$ for codes C_1 and C_2 and lower than or equal to $D_4 = \sqrt{5}D_1$ for code C_3, where D_1 is the minimum Euclidean distance between two constellation points.

Therefore, the basic idea behind the introduction of the symbol transformation is to artificially increase the Euclidean distance (ED) between these symbols for the second component code: T is designed in such a way that if the ED between symbols $s_{X_l^1}$ and $s_{X_l^2}$ is low, then the ED between $T(s_{X_l^1})$ and $T(s_{X_l^2})$ is high. In other words, if $s_{X_l^1}$ and

$s_{X_l^2}$ are mapped to points close to each other in the constellation, T is designed so that $T(s_{X_l^1})$ and $T(s_{X_l^2})$ are mapped to points far apart.

To this end, we have introduced the following parameter, $\delta(S_i, S_j)$, to measure the distance between two constellation points S_i and S_j, before and after transformation:

$$\delta(S_i, S_j) = d^2(S_i, S_j) + d^2(T(S_i), T(S_j)) \tag{4.2}$$

where $d^2(S_i, S_j)$ is the squared ED (SED) between points S_i and S_j in the constellation C. For a conventional 2-dimensional constellation, $d^2(S_i, S_j)$ is calculated as follows:

$$d^2(S_i, S_j) = (I_{S_i} - I_{S_j})^2 + (Q_{S_i} - Q_{S_j})^2 \tag{4.3}$$

where I and Q represent the *in-phase* and the *quadrature* axes for the QAM constellation respectively.

The proposed design criterion for the transformation T is to achieve the objective of maximizing parameter δ_{min} defined as:

$$\delta_{min} = \underbrace{min}_{(S_i, S_j) \in C^2} \delta(S_i, S_j) \tag{4.4}$$

Note that using this design criterion only requires the knowledge of the constellation. In particular, T does not depend on the TC interleaver.

When no transformation is used, δ_{min} is equal to $2D_1^2$, where D_1 is the minimum ED between two points in the constellation C.

For a constellation with q points, the total number of possible transformations is $q!$. For values of q greater than 16, an exhaustive search is hardly feasible in practice. Therefore, we have proposed a set of techniques that provide sub-optimal transformations but with values of δ_{min} much higher than $2D_1^2$. These techniques are described in the following sections.

4.3 Application example 1: NB-TC defined over GF(64) and mapped to a 64-QAM constellation

In the aim of defining the best possible transformation in the case of a 64-QAM constellation, the minimum value δ_{min} in Eq. (4.4) should be maximized. However, as already mentioned above, finding transformation with the highest value of δ_{min} amongst a total of 64! possibilities is not tractable using a brute force approach. Therefore, we first analyzed the distance distribution of the 64-QAM constellation and its geometrical features in order to find sub-optimal criteria for the design of good transformations without having to resort to an exhaustive search.

4.3.1 Distance distribution in a 64-QAM constellation

In a 64-QAM constellation, there are 33 non-zero different Euclidean distance values D_i between the constellation points. Each of these distances occurs $m(D_i)$ times. The values of D_i and $m(D_i)$ are provided in Table 4.1, where the distances are sorted in the ascending order ($D_{i+1} > D_i$). This table can be used to find some transformations that are able to increase the CSED of short DC sequences at least for one component code.

D_i^2	$m(D_i)$	D_i^2	$m(D_i)$	D_i^2	$m(D_i)$
$D_1^2=4$	112	$D_2^2=8$	98	$D_3^2=16$	96
$D_4^2=20$	168	$D_5^2=32$	72	$D_6^2=36$	80
$D_7^2=40$	140	$D_8^2=52$	120	$D_9^2=64$	64
$D_{10}^2=68$	112	$D_{11}^2=72$	50	$D_{12}^2=80$	96
$D_{13}^2=100$	128	$D_{14}^2=104$	84	$D_{15}^2=116$	72
$D_{16}^2=128$	32	$D_{17}^2=136$	60	$D_{18}^2=144$	32
$D_{19}^2=148$	56	$D_{20}^2=160$	48	$D_{21}^2=164$	48
$D_{22}^2=180$	40	$D_{23}^2=196$	16	$D_{24}^2=200$	46
$D_{25}^2=208$	32	$D_{26}^2=212$	24	$D_{27}^2=232$	20
$D_{28}^2=244$	24	$D_{29}^2=260$	16	$D_{30}^2=288$	8
$D_{31}^2=296$	12	$D_{32}^2=340$	8	$D_{33}^2=392$	2

Table 4.1: Euclidean distances D_i between couples of points in a 64-QAM constellation, with their corresponding multiplicity $m(D_i)$.

4.3.2 Unequal error protection in a 64-QAM constellation

The constellation points of a 64-QAM can be divided into 3 categories depending on their position in the constellation, as illustrated in Fig. 4.4. The coded symbols mapped to the points in the three sets differ in their number of neighbors in the constellation and therefore in their protection level. Corner symbols have only 3 direct neighbors: 2 at shortest distance D_1 and 1 at the (diagonal) second shortest distance $D_2 = \sqrt{2}D_1$. Remaining edge symbols (excluding the corner ones) have 5 direct neighboring symbols: 3 at the shortest distance D_1 and 2 at the distance D_2, while the inner symbols of the constellation have 4 neighbors at the shortest distance D_1 and 4 at the second shortest distance D_2. Consequently, the 4 corner symbols enjoy the highest level of protection when compared to the other points of the constellation, the 24 edge symbols are less protected than the corner symbols but are better protected than the 36 inner symbols. This unequal protection against errors has a large impact on the code design, since for a given transmitted sequence, the number of low weight competing sequences depends on the position in the constellation of the transmitted symbols. For a transmitted sequence mainly containing corner and edge symbols, the number of competing sequences with low CSED will be lower than for transmitted sequences consisting of inner symbols and the resulting symbol error rate will also be lower. The presence of corners and edges in the constellation introduces non-linearity in the NB coded modulation scheme and the probability of errors of a transmitted sequence depends on the contents of this sequence.

The unequal error protection can be advantageously used by the transformation function T if it is designed in such a way that most of the inner symbols before (respectively after) transformation become corner or edge symbols after (respectively before) transformation. The aim is to make as many systematic symbols as possible in the transmitted sequence benefit from the effect of the constellation points with a higher protection at least for one component code.

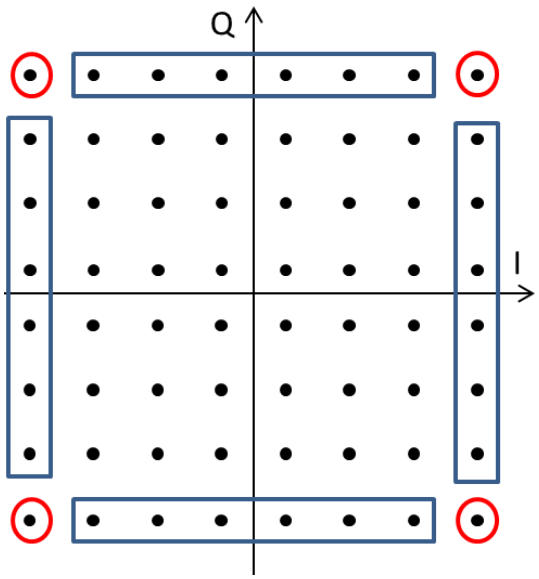

Figure 4.4: 64-QAM constellation with different protection levels for symbols.

4.3.3 Proposed algorithm

From the previous analysis, we added the following constraints for the search of good transformations:

- Transform systematic symbols mapped to neighboring points in the constellation into non-neighboring points with a guaranteed minimum ED separation. The aim is to avoid having low-CSED DC sequences for both component codes. We call this constraint *neighboring constraint*.

- Transform corner and edge symbols into inner symbols, in order to make as many systematic symbols as possible in the transmitted sequence benefit from the effect of the constellation points with a higher protection at least for one component code. We call this constraint *symbol protection leveling*.

Both constraints cause an exchange of the symbol protection levels between the first and the second encoder, thus decreasing the probability of occurrence of the same pattern of short DC sequences with a low CSED simultaneously for both component codes. Additionally, *symbol protection leveling* can reduce the number of low-CSED sequences competing with the transmitted sequence for one component code.

In practice, the algorithm used to find good transformations is described by the flow chart in Fig. 4.5. It operates on the $p = \frac{q(q-1)}{2}$ pairs of constellation points that are first sorted and listed in ascending order of their ED, according to Table 4.1: $L = (P_1, \ldots, P_i, \ldots, P_p)$, with $P_i = (S_{i,1}, S_{i,2})$ and $d(P_i) \leq d(P_{i+1})$. The couples are mapped step by step and the transformation is built accordingly. The basic principle is as follows: the algorithm steps through the list L in increasing order, starting with the pair with the lowest ED (P_1), and tries to assign its symbols, through the transformation, to the pair with the highest ED (thus, starting from the end of L), which satisfies the symbol protection leveling constraint (i.e. corner and edge points are mapped to inner points). Each time a symbol has been assigned a transformed symbol successfully, it is flagged in

all the pairs it belongs to and it does not need any further processing. The algorithm progresses until all the constellation symbols have been assigned successfully a transformed symbol.

During the progress of the algorithm, the transformation function is defined for each symbol in the constellation C separately. Therefore, in the intermediate steps processing the couples in the list L, three possibilities arise: both symbols in the couple are missing their transformation value, one symbol is still missing its transformation value or no symbol requires a transformation value any longer. Now, if r_i represents the number of symbols still missing their transformation values in each couple at index i ($i \in [1,p]$), we have $r_i \in [0,2]$ and $r_i = 2\ \forall i \in [1,p]$ at the beginning of the algorithm, since no transformation connection/value has been defined yet. We also use another variable l to go through L in decreasing order of the distances. At the beginning of the algorithm $l=p$, which corresponds to the total number of possible couples in the constellation.

The different steps of this algorithm are explained below:

- ***Step 1.*** **Find the lowest value of $m \in [1,p]$ such that $r_m \neq 0$**: in this step, we find the couple P_m, with at least one symbol still without a transformation value, having the lowest distance between its constituent symbols.

- ***Step 2.*** **Find the largest value of $n \in [1,l]$ such that $r_n = r_m$**: find $n \in [1,l]$, corresponding to the couple P_n with the largest distance between its symbols, and having the same number of non-assigned symbols as the couple P_m.

- ***Step 3. symbol protection leveling*** **constraint verification**: a test is performed in order to indicate whether the considered couples $(S_{m,1}, S_{m,2})$ and $(S_{n,1}, S_{n,2})$ respect the *symbol protection leveling* constraint. In other words, this test verifies that all the corner and edge symbols are transformed into inner symbols. When couples P_m and P_n respect this constraint, the algorithm goes to ***Step 4***. When this test fails, ***Step 2*** is run again in order to define a new value of $n_{\text{new}} \in [1,l]$, with $l = n_{\text{old}} - 1$ to limit the search ensemble and avoid retrieving the same couple $P_{n_{old}}$ again.

- ***Step 4.*** **Define new connections for T. Update $r_i\ \forall i \in [1,p]$**: in this step, one or two new connections are defined in the transformation block depending on the value of $r_m = r_n$, respecting all the constraints stated above and mapping the symbol(s) of P_m to the symbol(s) of P_n. Moreover, for all couples P_i, the value of r_i is updated if $S_{i,1}$ and/or $S_{i,2}$ is/are equal to $S_{m,1}$ and/or $S_{m,2}$.

- The algorithm iterates until assigning the q connections of the transformation.

According to Section 4.2, applying the *neighboring constraint* to avoid creating DC sequences with ED lower than d_2 for both component codes would ideally require mapping each couple of symbols whose distance is less than D_4 to a couple of symbols whose distance is greater than D_4 for code C_3 and mapping each couple of symbols whose distance is less than D_2 to a couple of symbols whose distance is greater than D_2 for codes C_1 and C_2.

In practice, as explained above, the conducted search was carried out independently of the code and we were able to find transformations verifying:

$$\begin{gathered} \forall (S_i, S_j) \in C^2 \\ \text{if } 0 < d(S_i, S_j) \leq D_3 \text{ then } d(T(S_i), T(S_j)) > D_3 \end{gathered} \tag{4.5}$$

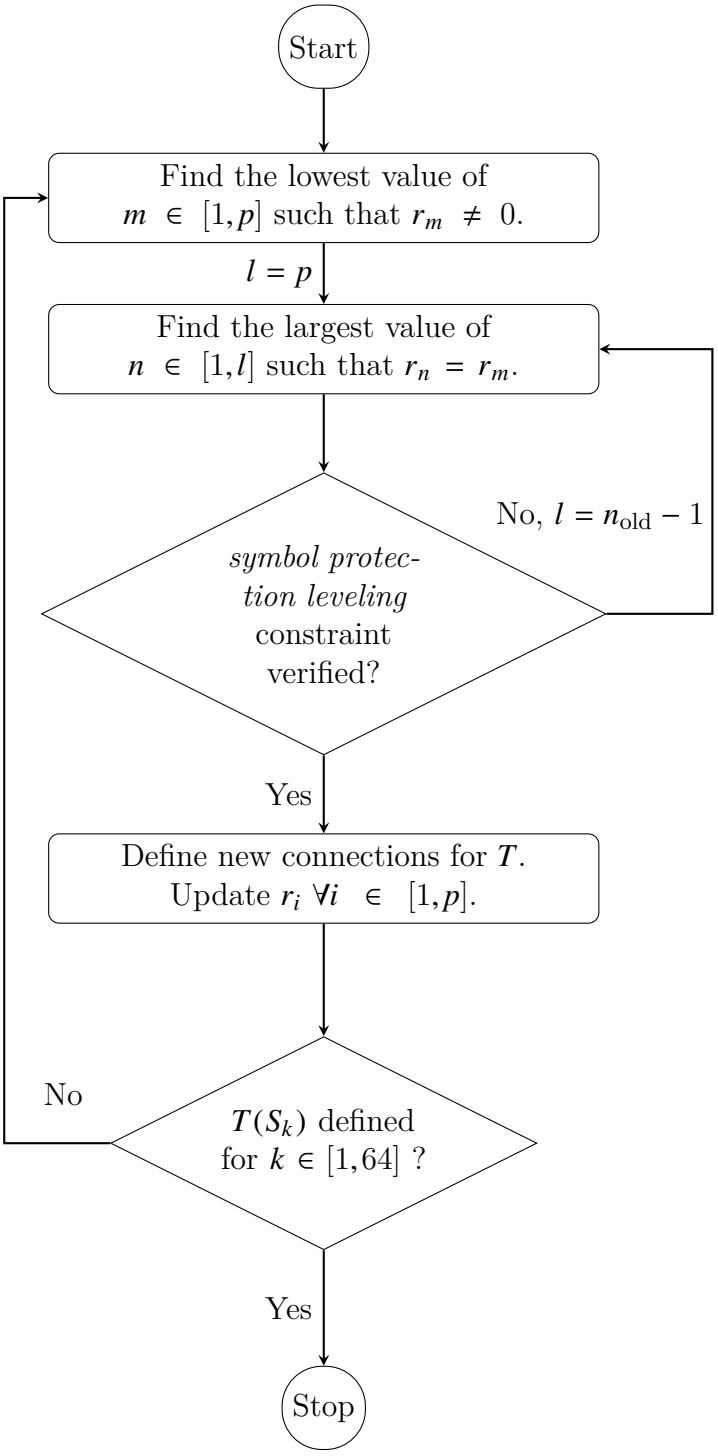

Figure 4.5: Flow chart of the proposed algorithm to find good transformations for a 64-QAM constellation.

but not for higher distance values. Therefore, it is not guaranteed that all the DC sequences have CSED less than d_2 for code C_3 are avoided at the second encoder. DC sequences having a couple (S_i, S_j) with $d(S_i, S_j) = D_4$ as competing systematic symbols are not totally avoided by the defined transformation. The number of these error-prone sequences is reduced thanks to the proposed algorithm, which defines a transformation connecting each of these couples to another with the highest possible distance between its constituent symbols. However, we expect it to improve the overall minimum CSED of the resulting TC.

4.4 Application example 2: NB-TC defined over GF(64) and mapped to a 4-QAM constellation

When mapped to a 4-QAM constellation, each symbol $S_i \in$ GF(64) is mapped into three 4-QAM symbols s_i^1, s_i^2 and s_i^3. The design criterion of the transformation remains the maximization of δ_{min} from Eq. (4.4), in order to avoid having the same competing symbols at both encoders. However, in this case, the lowest non-zero Euclidean distance between two GF(64) symbols S_i and S_j is obtained when two of their constituent 4-QAM symbols are identical (for example, when $s_i^1 = s_j^1$ and $s_i^2 = s_j^2$). To this end, in order to maximize δ_{min} in Eq. (4.4), two constraints are to be satisfied:

- Transform all couples of GF(64) symbols (S_i, S_j) having two identical constituent 4-QAM symbols into GF(64) couples $(T(S_i), T(S_j))$ having no common 4-QAM symbols.
- Transform the maximum number of GF(64) couples (S_i, S_j) having one common constituent 4-QAM symbol into GF(64) couples $(T(S_i), T(S_j))$ having no common 4-QAM symbols.

Due to these new constraints, modifications should be introduced into the proposed algorithm of Fig. 4.5 as shown in Fig. 4.6.

The Euclidean distance between two GF(64) symbols (S_i, S_j) mapped to a 4-QAM constellation is given by:

$$\begin{aligned} d^2(S_i, S_j) = &(I_{s_i^1} - I_{s_j^1})^2 + (Q_{s_i^1} - Q_{s_j^1})^2 \\ &+(I_{s_i^2} - I_{s_j^2})^2 + (Q_{s_i^2} - Q_{s_j^2})^2 \\ &+(I_{s_i^3} - I_{s_j^3})^2 + (Q_{s_i^3} - Q_{s_j^3})^2 \end{aligned} \tag{4.6}$$

where I and Q represent the in-phase and quadrature axes of the 4-QAM constellation.

Except for the third step, the modified algorithm applies the same steps as the ones explained in section 4.3 and shown in Fig. 4.5. The modified step 3 of Fig. 4.6 now verifies if all the couples of GF(64) symbols that are composed of two identical 4-QAM symbols are mapped to couples not having identical 4-QAM symbols after transformation. $Iden_i$ defines the number of identical 4-QAM symbols between the GF(64) symbols in couple P_i of the list L, sorted in ascending order of their ED computed by Eq. (4.6). When $Iden_m = 2$ and $Iden_n > 0$, the symbols of P_m share two identical 4-QAM symbols, while the symbols of P_n share one or more identical 4-QAM symbols. In this last case, the proposed constraint is not verified, and a new couple P_n should be found in L. Otherwise, the algorithm can continue to ***Step 4***.

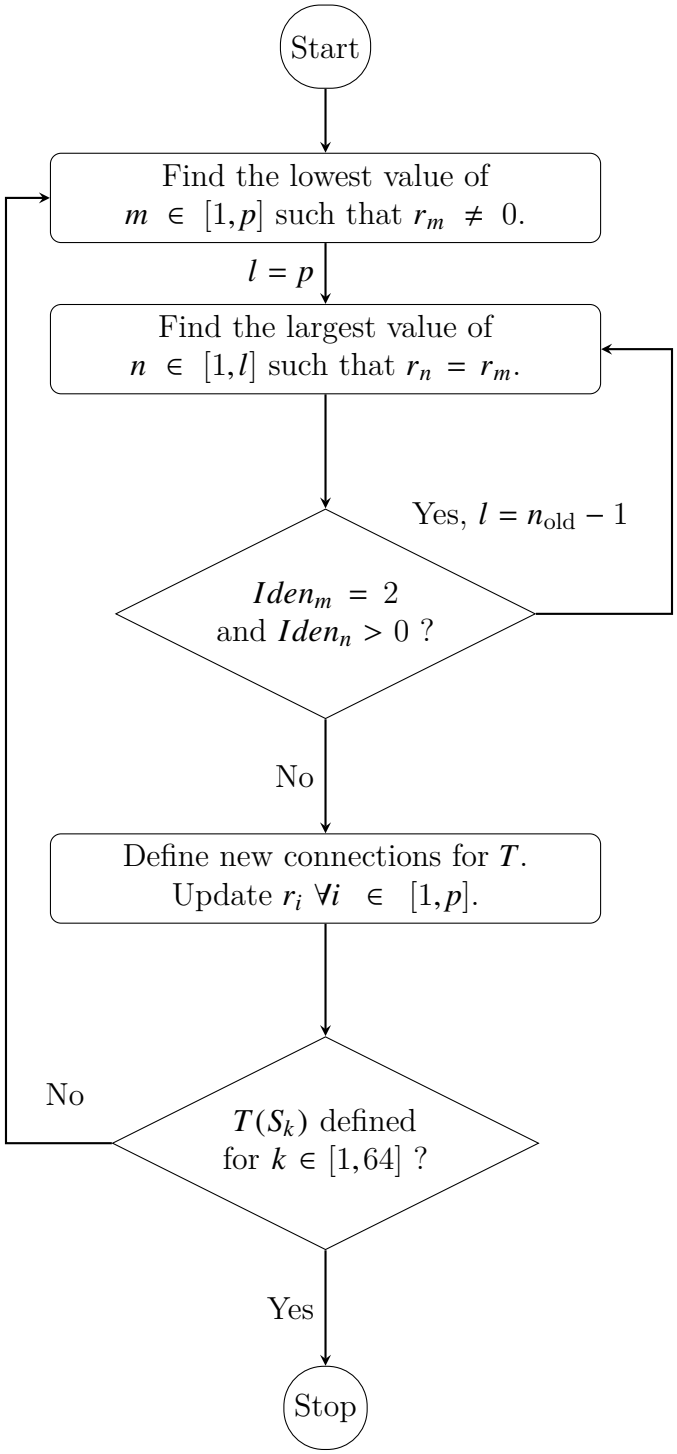

Figure 4.6: Flow chart of the proposed algorithm to find good transformations for a 4-QAM constellation.

4.5 Results and discussion

4.5.1 Application example 1: NB-TC defined over GF(64) and mapped to a 64-QAM constellation

We ran the algorithm described in Section 4.3 with a 64-QAM constellation respecting the mapping of Table 4.2, with $b_5b_4b_3b_2b_1b_0$ denoting the binary image of the constellation symbols, where b_5 corresponds to the most significant bit of the symbol representation.

Q value	$b_5b_3b_1$	I value	$b_4b_2b_0$
+ 7	000	+ 7	000
+5	001	+5	001
+3	011	+3	011
+1	010	+1	010
-1	110	-1	110
-3	111	-3	111
-5	101	-5	101
-7	100	-7	100

Table 4.2: Binary mapping of the in-phase I and quadrature Q axes for 64-QAM.

x	$T(x)$	x	$T(x)$	x	$T(x)$	x	$T(x)$
0	15	16	7	32	11	48	41
1	19	17	25	33	47	49	62
2	29	18	39	34	28	50	31
3	0	19	9	35	50	51	33
4	3	20	51	36	63	52	45
5	35	21	46	37	27	53	22
6	12	22	37	38	10	54	49
7	4	23	17	39	5	55	18
8	6	24	14	40	57	56	43
9	8	25	2	41	44	57	60
10	38	26	59	42	54	58	55
11	32	27	52	43	21	59	56
12	1	28	34	44	58	60	13
13	16	29	26	45	53	61	20
14	23	30	61	46	30	62	36
15	24	31	48	47	42	63	40

Table 4.3: The best transformation found for a 64-QAM constellation.

We obtained the transformation $T(x)$ shown in Table 4.3, with $\delta_{min} = 28$. Note that several equivalent transformations can be obtained, depending on the respective locations of symbol pairs with the same distances in list L.

Table 4.4 shows the distribution of the distances, after transformation, between the symbols whose distance before transformation is lower than or equal to D_3. One can observe, as mentioned in Section 4.3, that they are all greater than or equal to D_4. Furthermore, the distance spectrum after transformation for couples having $d = D_4$ before transformation is shown in Table 4.5. Respecting the proposed algorithm, these spectra are optimized in terms of connecting each pair of constellation points with low distance to a one with the highest possible distance.

D_i	$m(D_i)$	D_i	$m(D_i)$	D_i	$m(D_i)$
D_1	0	D_2	0	D_3	0
D_4	25	D_5	16	D_6	21
D_7	27	D_8	19	D_9	10
D_{10}	22	D_{11}	13	D_{12}	15
D_{13}	29	D_{14}	6	D_{15}	7
D_{16}	5	D_{17}	12	D_{18}	3
D_{19}	8	D_{20}	5	D_{21}	3
D_{22}	9	D_{23}	5	D_{24}	12
D_{25}	7	D_{26}	3	D_{27}	8
D_{28}	4	D_{29}	6	D_{30}	2
D_{31}	1	D_{32}	2	D_{33}	1

Table 4.4: Distribution of distances $d(T(S_1),T(S_2))$, after transformation, for couples (S_1, S_2) having $d(S_1, S_2) \leq D_3$ before transformation.

D_i	$m(D_i)$	D_i	$m(D_i)$	D_i	$m(D_i)$
D_1	0	D_2	11	D_3	19
D_4	21	D_5	5	D_6	3
D_7	6	D_8	5	D_9	1
D_{10}	10	D_{11}	5	D_{12}	9
D_{13}	11	D_{14}	7	D_{15}	8
D_{16}	3	D_{17}	4	D_{18}	3
D_{19}	5	D_{20}	5	D_{21}	5
D_{22}	4	D_{23}	2	D_{24}	8
D_{25}	1	D_{26}	3	D_{27}	2
D_{28}	0	D_{29}	0	D_{30}	1
D_{31}	0	D_{32}	1	D_{33}	0

Table 4.5: Distribution of distances $d(T(S_1),T(S_2))$, after transformation, for couples (S_1, S_2) having $d(S_1, S_2) = D_4$ before transformation.

An example of DC sequences with low CSED before transformation is depicted in Fig. 4.7 for a TC using C_1 as a constituent encoder. Suppose that input symbols 45 and 47 are part of the transmitted data. The lowest distance between the competing DC sequences is observed when symbols 12 and 41 create a competing DC sequence with the

transmitted one, diverging together from state 6 and converging together to the same state.

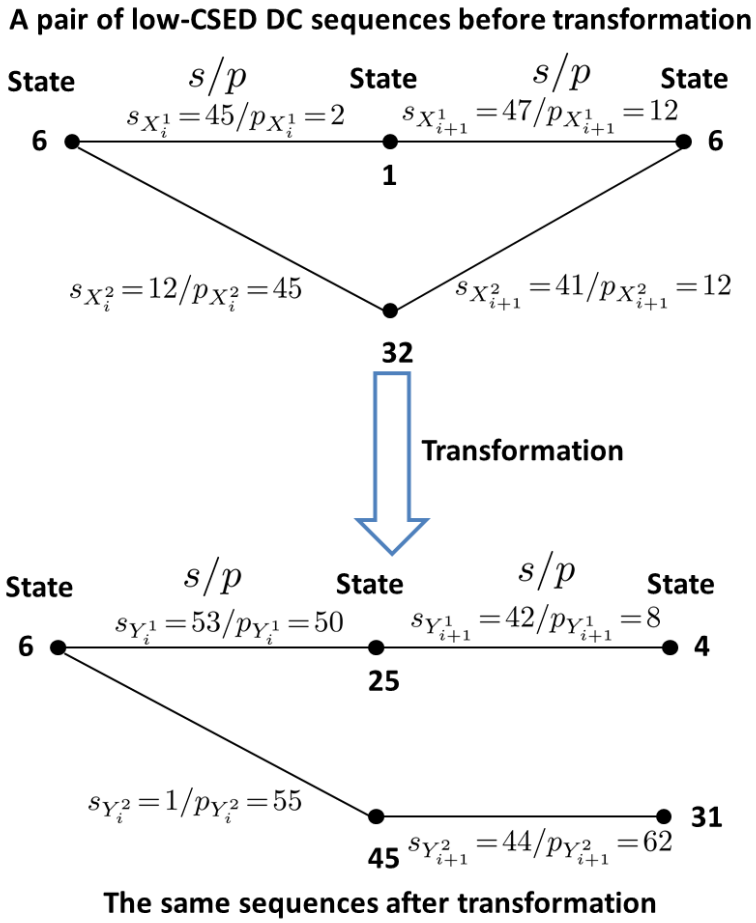

Figure 4.7: An example of short DC sequences before and after transformation, for code C_1 in Table 2.4.

This transmitted sequence is particularly prone to channel errors since the constellation points labeled by $s_{X_i^1} = 45$ and $s_{X_i^2} = 12$ on one hand and $s_{X_{i+1}^1} = 47$ and $s_{X_{i+1}^2} = 41$ on the other hand are neighbors in the constellation (according to the used mapping, $d(45,12) = d(47,41) = D_2$). Additional discrimination between these two sequences resides in the different parity symbol values along the DC trellis paths. But these symbols only differ in $p_{X_i^1} = 2$ and $p_{X_i^2} = 45$ since the second parity symbol is identical in both sequences.

By applying the transformation depicted in Table 4.3, Fig. 4.7 shows that symbols $s_{X_i^1} = 45$ and $s_{X_{i+1}^1} = 47$ are transformed into symbols $s_{Y_i^1} = 53$ and $s_{Y_{i+1}^1} = 42$ at the second encoder. Then, the transformed symbols do not form a pair of length-2 DC sequences any longer. Before transformation, the CSED between these competing sequences is: $d^2_{\text{RSC1}} = d^2(12,45) + d^2(2,45) + d^2(47,41) = D_2^2 + D_8^2 + D_2^2$, according to Table 4.1 $d^2_{\text{RSC1}} = 68$. After transformation, $d^2_{\text{RSC2}} = d^2(1,53) + d^2(50,55) + d^2(42,44) + d^2(8,62) = D_{29}^2 + D_3^2 + D_7^2 + D_{12}^2 = 396$. The CSED between these sequences is therefore increased thanks to the transformation.

We first conducted a performance evaluation of the designed transformation through simulations over an Additive White Gaussian Noise (AWGN) channel for NB-TCs over

GF(64), with uniform interleaving and a sequence length $K_s = 900$ GF(64) symbols. Simulations were performed with and without applying the transformation. NB-CCs C_1 and C_3 described in Table 2.4 were used since they represent the codes with the worst and best achieved CSED spectrum, respectively. Any other code should provide results lying in between these two cases. Fig. 4.8 shows that, for both codes, a gain of more than one decade in terms of FER is achieved in the error floor region for both codes.

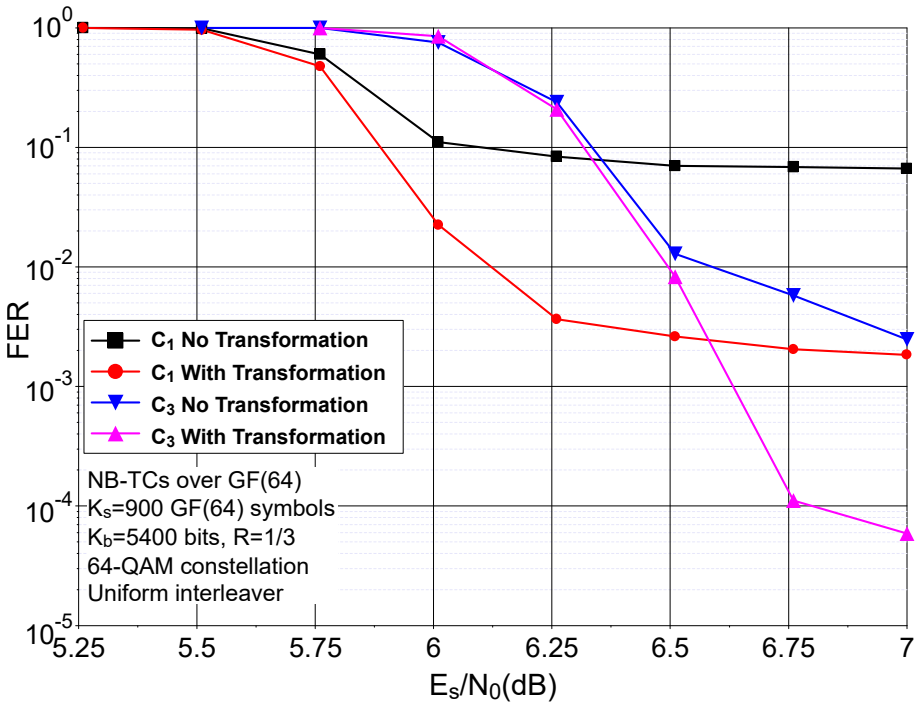

Figure 4.8: Frame error rate performance comparison with uniform interleavers over an AWGN channel with 64-QAM modulation, $K_s = 900$ GF(64) symbols and $R = 1/3$ for C_1 and C_3 in GF(64) with and without symbol transformation.

Moreover, simulations were also carried out with an ARP interleaver [60], in order to test the effect of the transformation when used with a practical structured interleaver. The results are presented in Fig. 4.9 for $K_s = 160$ GF(64) symbols. The used ARP interleaver was designed respecting the constraints stated in Chapter 3 and its parameters are presented in Table 4.6. Truncated Union Bounds (TUB) were calculated following the method explained in Chapter 3 and are depicted with dashed lines in the figure. The obtained FER and TUB curves show that higher gains, more than two decades in terms of FER, can be achieved in the error floor region for both codes, when a structured interleaver is used. For the NB-TC using C_3, the error floor is predicted to occur in the order of 10^{-10} of FER when the transformation is inserted into the code structure.

S_{min}	Girth	P	Q	(S(0),...,S(Q-1))
8	8	143	8	(0,150,62,10,110,22,20,98)

Table 4.6: ARP interleaver parameters for the NB-TCs over GF(64) simulated in Fig. 4.9, $R = 1/3$, $K_s = 160$ GF(64) symbols. ARP equation: $\Pi(i) = (Pi + S\,(i \bmod Q)) \bmod K_s$.

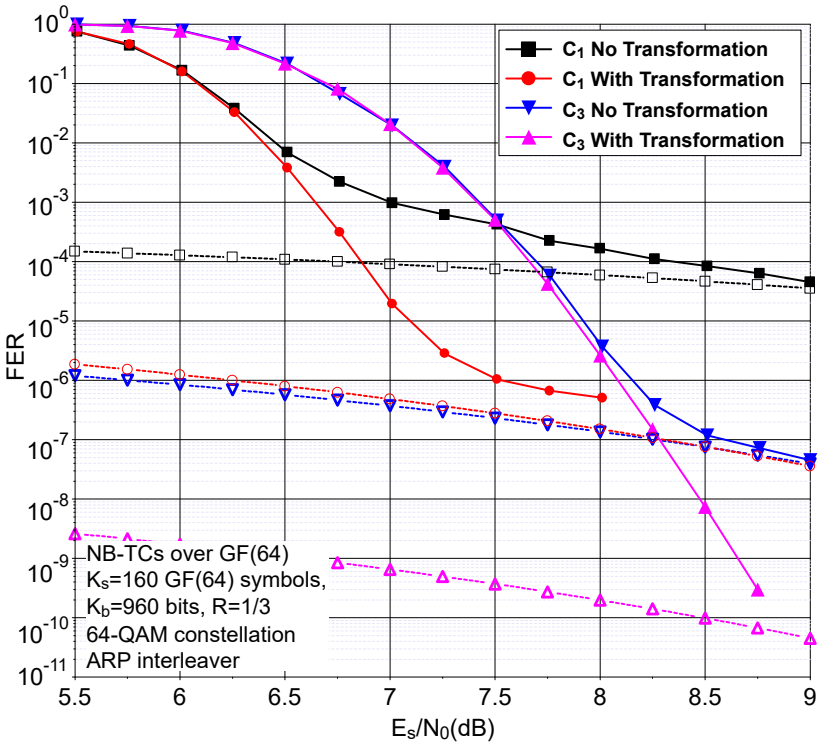

Figure 4.9: Frame error rate performance comparison with ARP interleaver over an AWGN channel with 64-QAM modulation, $K_s = 160$ GF(64) symbols and $R = 1/3$ for C_1 and C_3 in GF(64) with and without symbol transformation.

4.5.2 Application example 2: NB-TC defined over GF(64) and mapped to a 4-QAM constellation

When considering NB-TCs over GF(64), each resulting symbol S_i is mapped to 3 different 4-QAM points, denoted by s_i^1, s_i^2 and s_i^3. Let $b_5b_4b_3b_2b_1b_0$ be the binary image of S_i: s_i^1 is constructed from the 2 MSB bits b_5b_4, s_i^2 is constructed from b_3b_2, and s_i^3 from b_1b_0. The used 4-QAM constellation respects the mapping of Table 4.7, with $b'_1b'_0$ denoting the binary image of the constellation symbols, where b'_1 corresponds to the most significant bit of the symbol representation.

Q value	b'_1	I value	b'_0
+1	0	+1	0
-1	1	-1	1

Table 4.7: Binary mapping of the in-phase I and quadrature Q axes for the 4-QAM.

The best transformation $T(x)$ obtained for a 4-QAM constellation, respecting the constraints stated in Section 4.4 is given in Table 4.8. The minimum value of the parameter δ defined in Eq. (4.2) is $\delta_{min} = 16$.

Similarly to what is shown in Section 4.5.1, an example of a couple of competing sequences is shown in Fig. 4.10. The considered sequences diverge from state 5 and converge to state 32 after two trellis stages at the first encoder. This couple of sequences shows a low CSED at the first encoder, since the first couple of competing systematic

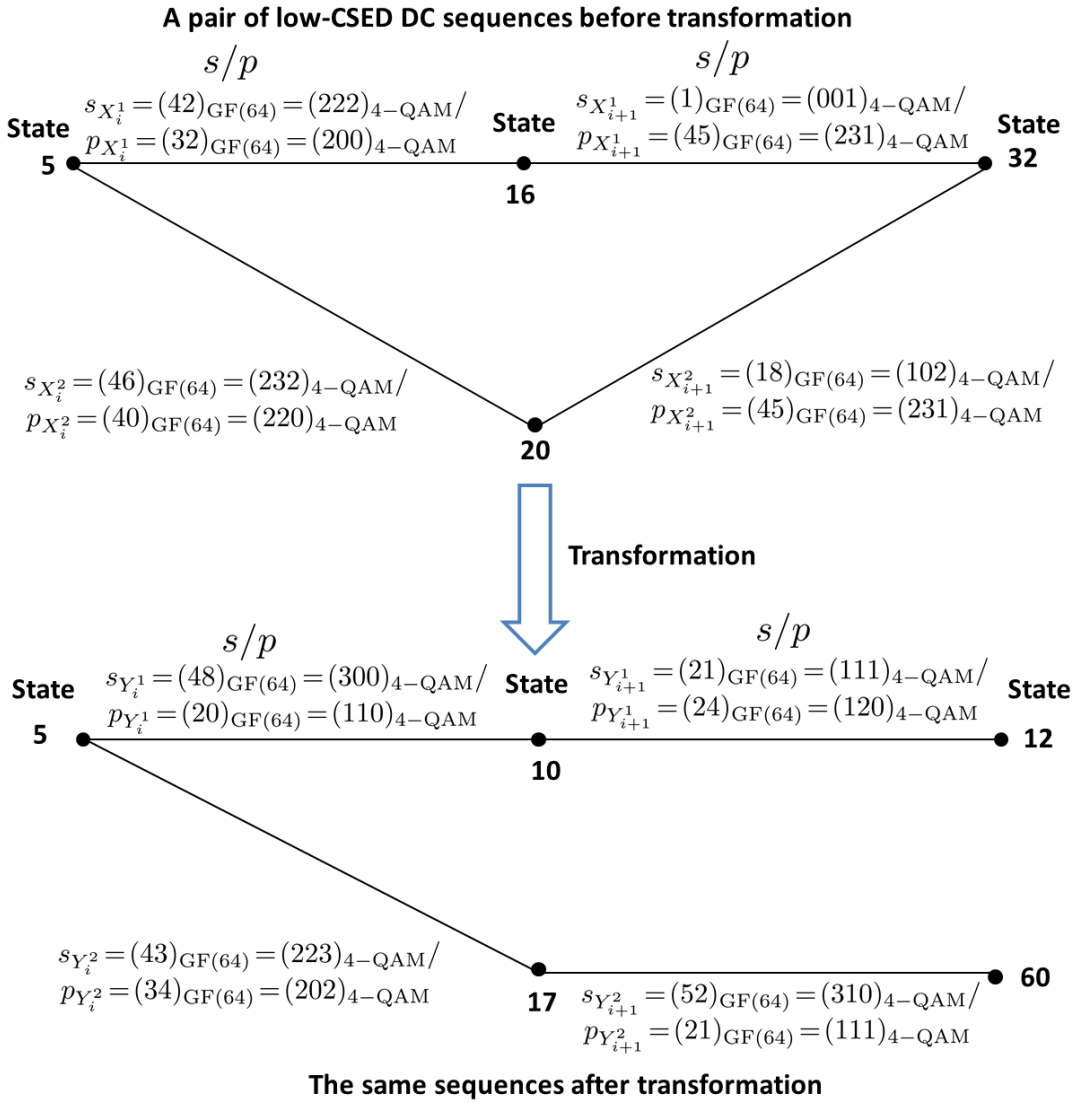

Figure 4.10: An example of short DC sequences before and after transformation, for code C_1 in Table 2.4.

x	$T(x)$	x	$T(x)$	x	$T(x)$	x	$T(x)$
0	0	16	30	32	39	48	57
1	21	17	11	33	50	49	44
2	42	18	52	34	13	50	19
3	63	19	33	35	24	51	6
4	27	20	5	36	60	52	34
5	14	21	16	37	41	53	55
6	49	22	47	38	22	54	8
7	36	23	58	39	3	55	29
8	45	24	51	40	10	56	20
9	56	25	38	41	31	57	1
10	7	26	25	42	48	58	46
11	18	27	12	43	37	59	59
12	54	28	40	44	17	60	15
13	35	29	61	45	4	61	26
14	28	30	2	46	43	62	53
15	9	31	23	47	62	63	32

Table 4.8: The best transformation found for a 4-QAM constellation.

symbols ($s_{X_i^1} = 42$ and $s_{X_i^2} = 46$) share two common 4-QAM symbols, and then $d^2(42,46)$ is low. In addition, the second systematic couple ($s_{X_{i+1}^1} = 1$ and $s_{X_{i+1}^2} = 18$) shows a SED $d^2(1,18)$ not excessively high since its constituent symbols share one 4-QAM symbol. After transformation, it is shown that the couple ($s_{X_i^1} = 42$ and $s_{X_i^2} = 46$) is transformed to ($s_{Y_i^1} = 48$ and $s_{Y_i^2} = 43$) following Table 4.8. The new defined couple at the second encoder shows a high value of SED $d^2(48,43)$ since ($s_{Y_i^1} = 48$ and $s_{Y_i^2} = 43$) do not share any 4-QAM symbol. In conclusion, these competing DC sequences at the first encoder are transformed into non-DC sequences at the second encoder, and their CSED has largely increased.

To assess the proposed transformation, simulations are performed over an AWGN channel, comparing the performance of NB-TC over a 4-QAM constellation, and using uniform interleavers for $K_s = 900$ GF(64) symbols. Fig. 4.11 shows the comparison of results for C_1 and C_3. Similarly to the case of a 64-QAM, a gain of more than a decade in the error floor region is observed for the two considered codes when using the transformation.

A comparison is also carried out when using an ARP interleaver. In this example, only NB-TCs using constituent code C_1 are simulated in order to be able to observe the error floor region, since the error floors obtained with C_3 and an ARP interleaver are well below our software simulation capabilities. The data block size is $K_s = 30$ GF(64) symbols, and the ARP interleaver respects the parameters shown in Table 4.9. Results are shown in Fig. 4.12, where the NB-TC is shown to perform better when a transformation block is applied. A gain of more than two decades in terms of FER is observed in the error floor region when using a transformation.

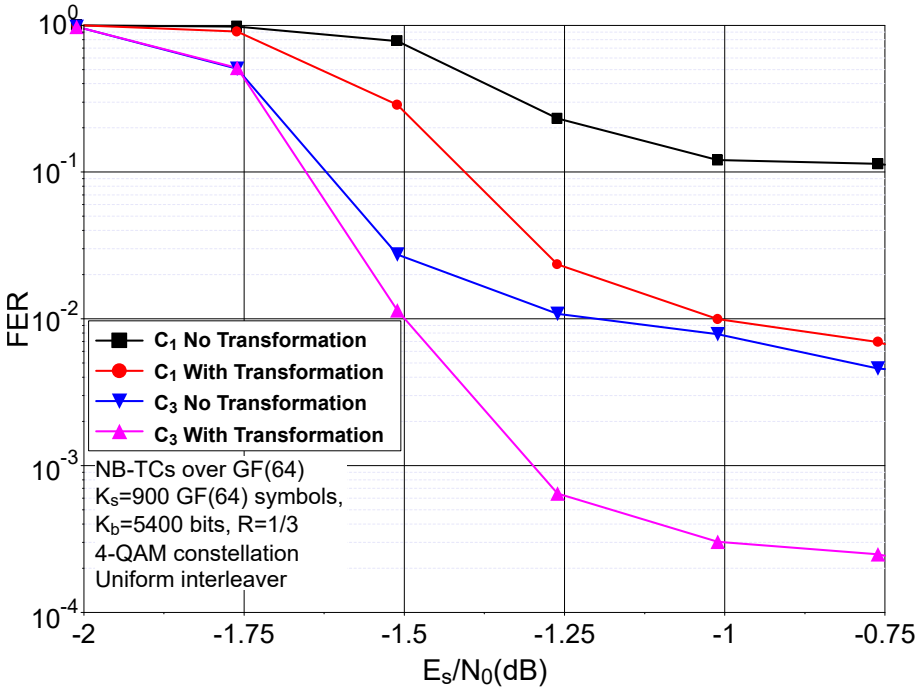

Figure 4.11: Frame error rate performance comparison with uniform interleavers over an AWGN channel with a 4-QAM constellation, K_s = 900 GF(64) symbols and R = 1/3 for C_1 and C_3 in GF(64) with and without symbol transformation.

S_{min}	Girth	P	Q	(S(0),...,S(Q-1))
6	4	19	3	(0,25,14)

Table 4.9: ARP interleaver parameters for the NB-TCs over GF(64) simulated in Fig. 4.12, R = 1/3, K_s = 30 GF(64) symbols. ARP equation: $\Pi(i) = (Pi + S(i \bmod Q)) \bmod K_s$.

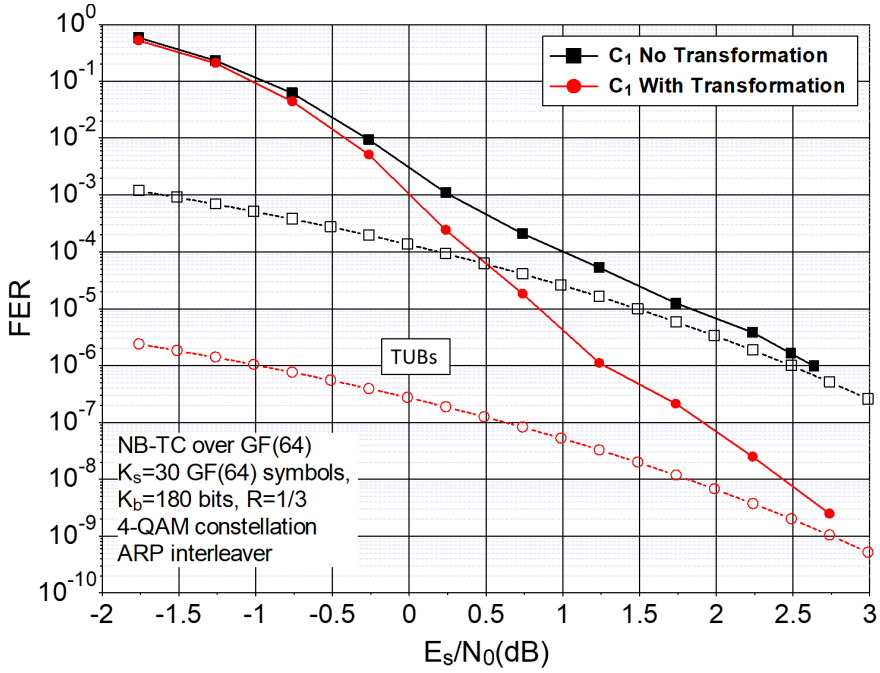

Figure 4.12: Frame error rate performance comparison with ARP interleaver over an AWGN channel with 4-QAM modulation, $K_s = 30$ GF(64) symbols and $R = 1/3$ for C_1 in GF(64) with and without symbol transformation.

4.6 Error correction performance of NB-TCs and comparison with state-of-the-art codes

In this section, we compare the performance of NB-TCs for different simulation scenarios. First, we begin by comparing NB-TCs with existing binary TCs, which are divided into two categories: binary TCs adopted in the LTE standard [29], and the enhanced binary TCs described in [89, 13]. Second, we compare them with the binary LDPC codes adopted in the fifth generation of mobile communication (5G) [93]. Additionally, comparisons are conducted with NB-LDPC codes for different scenarios. Then, we show multiple simulation comparisons between our NB-TCs and the polar codes adopted in the 5G standard [93]. Finally, it is shown that NB-TCs can outperform the BICM capacity for high-order modulation.

4.6.1 Comparison with binary turbo codes

As a first step for assessing the proposed NB-TCs, we compare their error correction performance with the binary TCs. Two different families of binary TCs are considered: the TCs adopted in the LTE standard [29], and the enhanced binary TCs [89, 13].

4.6.1.1 Comparison with LTE TCs

Fig. 4.13 compares the designed NB-TCs and the binary LTE code for a transmission over an AWGN channel with a 4-QAM constellation, and for a frame size $K_s = 180$ GF(64) symbols or $K_b = 1080$ bits. Two coding rates are considered: $R = 1/3$ and $R = 9/10$. Interleaving and puncturing for the NB-TCs respect the constraints presented in Chapter

3. The ARP interleaver parameters are shown in Table 4.10, and the puncturing mask for $R = 9/10$ is presented in Table 4.11, with s being the systematic part of the codeword, p_1 and p_2 being the parity positions of the first and second encoder, respectively. NB-TCs are shown to outperform the LTE code at high and low error rates for both scenarios.

P	Q	(S(0),...,S(Q-1))
7	3	(0,73,134)

Table 4.10: ARP interleaver parameters for K_s = 180 GF(64) symbols (ARP equation: $\Pi(i) = (Pi + S(i \bmod Q)) \bmod K_s$).

s	p_1	p_2
111111111	000000000	100000000
111111110	000000001	010000000

Table 4.11: Puncturing mask for the NB-TC with R = 9/10, and K_s = 180 GF(64) symbols. The puncturing period is $M = 18$.

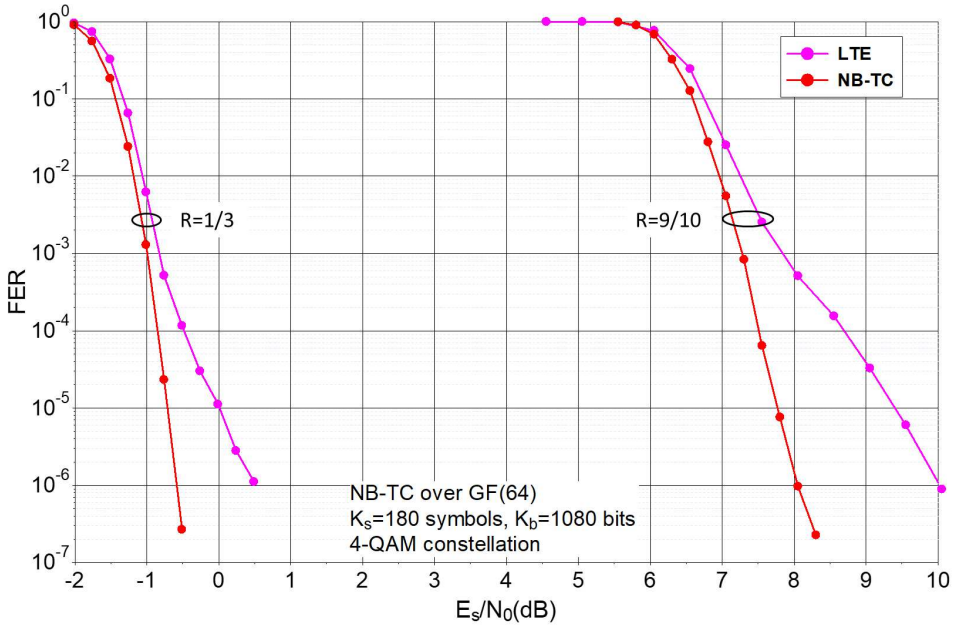

Figure 4.13: Frame error rate performance comparison of NB-TC over GF(64), $K_s = 180$ GF(64) symbols, and LTE binary TC, K_b = 1080 bits, AWGN channel and 4-QAM constellation.

In Fig. 4.14, another comparison with the LTE code is shown for a transmission using a 64-QAM constellation. $K_s = 30$ GF(64) symbols, $K_b = 180$ bits, and three coding rates are considered. The worst (C_1) and best (C_3) NB-CCs are used as constituent codes for the NB-TCs. The ARP interleaver parameters are given in Table 4.12. Bit puncturing is considered to achieve $R = 1/2$ (see Section 3.2.2), and the symbol puncturing mask for rate $R = 2/3$ is shown in Table 4.13. C_1 is shown to achieve better convergence than the LTE

codes, while performing at a similar error floors. C_3 offers better to similar convergence while largely lowering the error floor.

P	Q	(S(0),...,S(Q-1))
19	3	(0,25,14)

Table 4.12: ARP interleaver parameters for $K_s = 30$ GF(64) symbols.

s	p_1	p_2
1111111111	0010001000	0100010100

Table 4.13: Puncturing mask for the NB-TC with $R = 2/3$, and $K_s = 30$ GF(64) symbols. The puncturing period is $M = 10$.

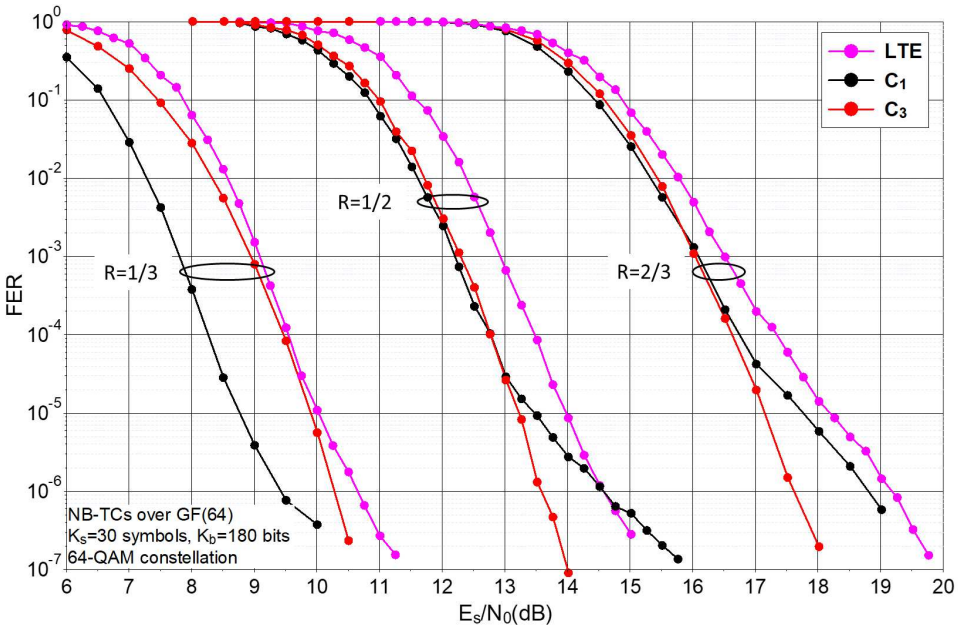

Figure 4.14: Frame error rate performance comparison of NB-TC over GF(64), $K_s = 30$ GF(64) symbols, and LTE binary TC, $K_b = 180$ bits, AWGN channel and 64-QAM constellation.

In conclusion, NB-TCs are shown to outperform the binary LTE code for different modulation orders, information frame sizes, and coding rates. This gain can be seen at high and low error rates, where NB-TCs are shown to be able to remedy the high error floor problem of the LTE codes.

4.6.1.2 Comparison with enhanced binary TCs

The performance of binary TCs was enhanced by proposing a joint optimization of the permutation functions and the puncturing patterns [89, 13]. NB-TCs are compared with these codes under different simulation conditions. First, a small frame size was considered

over a 4-QAM constellation. Binary and NB-TCs were designed to respect the interleaving and puncturing constraints stated in Chapter 3. K_s is equal to 30 symbols in the case of NB-TCs, and the interleaver respects the parameters of Table 4.12. The puncturing for the case of NB-TCs complies with the mask of Table 4.13. Binary TCs are designed with $K_b = 208$ bits, the interleaver is designed using the parameters of Table 4.14, and the puncturing respects the mask of Table 4.15. When comparing the performance of these codes, Fig. 4.15 shows that even with a slightly shorter frame size, NB-TCs outperform binary TCs for different coding rates. This gain is mainly in the error floor region, which is improved for NB codes thanks to the lower experienced correlation at the decoder end.

P	Q	(S(0),...,S(Q-1))
159	16	(8, 96, 23, 42, 170, 103, 202, 10, 200, 21, 24, 88, 5, 152, 136, 96)

Table 4.14: ARP interleaver parameters for $K_b = 208$ bits.

s	p_1	p_2
11111111	10010000	00100100
11111111	10000100	01000010

Table 4.15: Puncturing mask for binary TC with $R = 2/3$, and $K_b = 208$ bits. The puncturing period is $M = 16$.

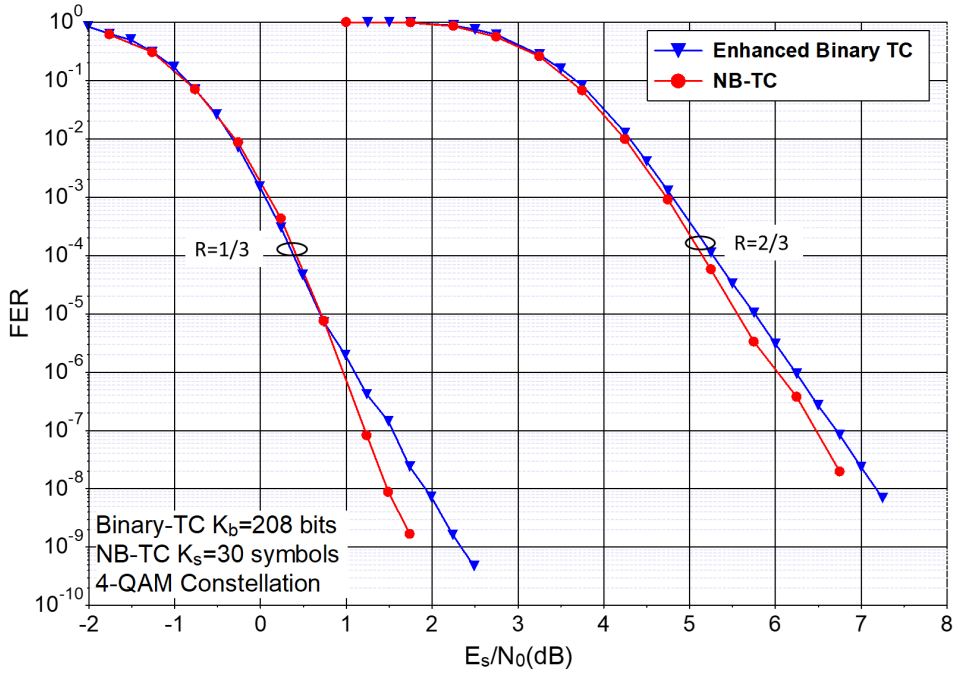

Figure 4.15: Frame error rate performance comparison of NB-TC over GF(64) and enhanced binary TC, $K_s = 30$ GF(64) symbols for the NB-TC and $K_b = 208$ bits for the binary TC, AWGN channel and 4-QAM constellation.

A second comparison is conducted, where a 64-QAM constellation is considered. In this context, NB codes are expected to achieve a higher correction capability thanks to the observed capacity gap between CM and BICM. The data frame sizes are $K_s = 900$ GF(64) symbols (equivalent to $K_b = 5400$ bits) for the case of NB-TCs, and $K_b = 6000$ bits in the case of binary TCs. The adopted interleaver for the NB-TCs is an ARP interleaver, whose parameters are shown in Table 4.16, while bit puncturing is adopted to achieve $R = 1/2$ (see Section 3.2.2). In this context, as shown in Fig. 4.16, the NB-TC outperforms the binary code by more than 0.5 dB for the considered coding rates, despite a higher frame size in the case of binary TC. This gain is in accordance with the gains predicted by the CM and BICM capacity curves in Section 1.2.3.2 of Chapter 1.

P	Q	(S(0),...,S(Q-1))
137	4	(0,854,396,362)

Table 4.16: ARP interleaver parameters for $K_s = 900$ GF(64) symbols.

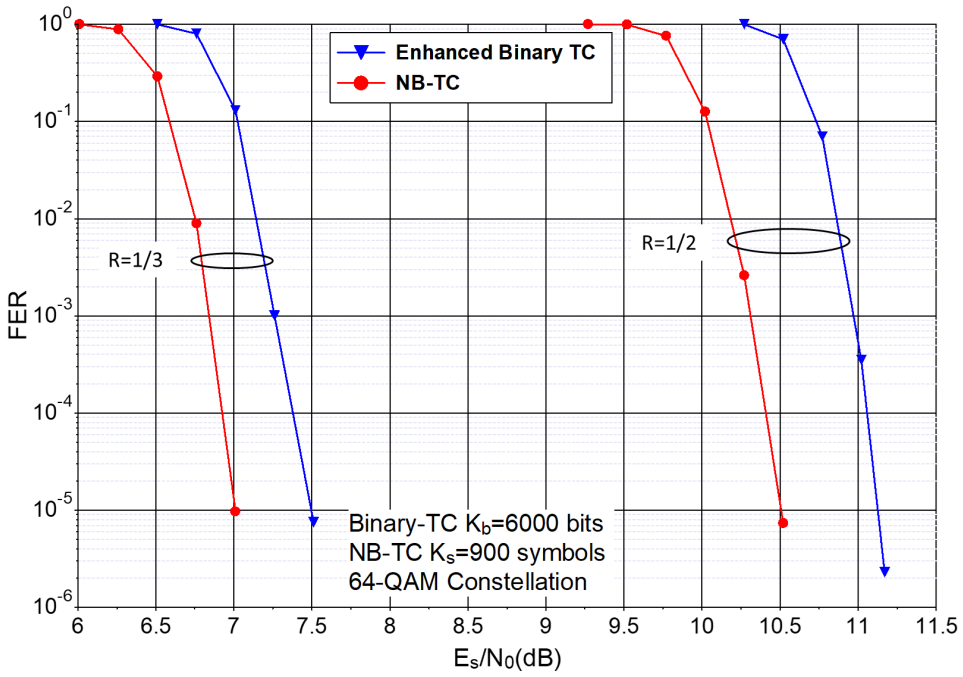

Figure 4.16: Frame error rate performance comparison of NB-TC over GF(64) and enhanced binary TC, $K_s = 900$ GF(64) symbols for the NB-TC and $K_b = 6000$ bits for the binary TC, AWGN channel and 64-QAM constellation.

4.6.2 Comparison with binary LDPC codes

Two binary LDPC codes were adopted for the data channel communication in the 3GPP specification for the 5G New Radio (NR) [93]. The advantages of these codes are the following [94]: mainly they can offer better area efficiency than the LTE TC, possible decoding parallelization and improved latency, and also better performance is guaranteed for different coding rates and frame sizes.

Several comparisons of NB-TCs and 5G-LDPC codes were conducted. The LDPC codes were simulated with the Matlab code of [95], using 50 iterations of the BP decoding algorithm.

The first comparison considers a transmission using a 4-QAM constellation with $R = 9/10$ and $K_b = 1080$ bits. NB-TCs are designed with $K_s = 180$ GF(64) symbols. The interleaver parameters and puncturing patterns are given in Tables 4.10 and 4.11, respectively. Fig. 4.17 shows that, at high error rates, NB-TCs slightly outperform the 5G-LDPC codes, while they largely lower the error floor.

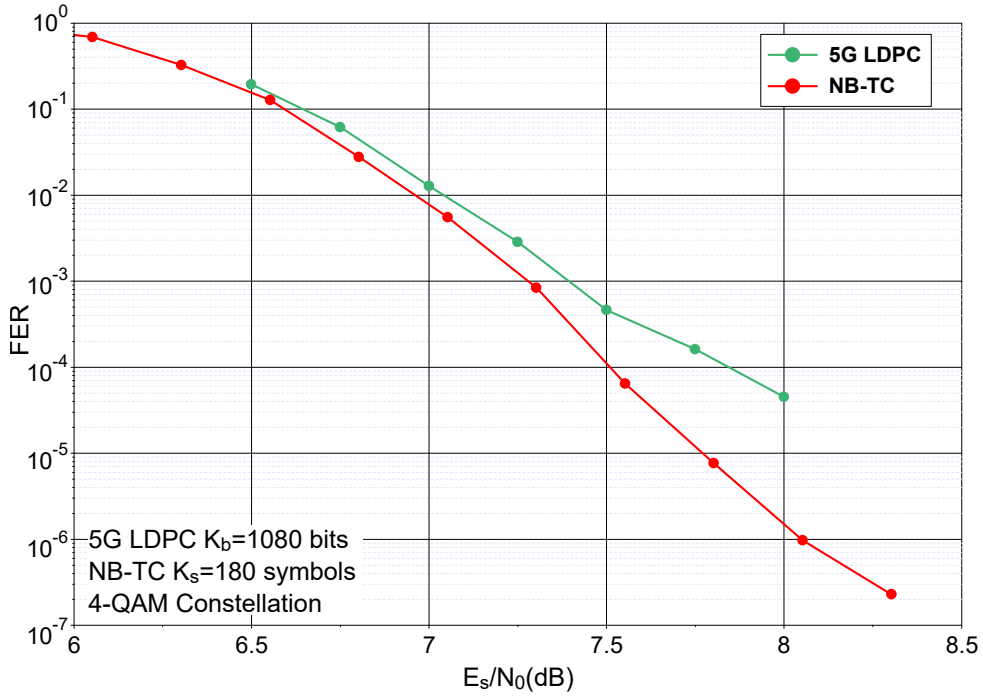

Figure 4.17: Frame error rate performance comparison of NB-TC defined over GF(64), $K_s = 180$ GF(64) symbols, and 5G-LDPC code, $K_b = 1080$ bits, $R = 9/10$, AWGN channel and 4-QAM constellation.

A second set of comparisons is conducted, with a 64-QAM constellation. Two scenarios are considered: $K_b = 960$ bits/$K_s = 160$ GF(64) symbols with $R = 1/3$ and $K_b = 5400$ bits/$K_s = 900$ GF(64) symbols with $R = 1/2$. Two NB-TCs, with different component codes, are assessed when considering $K_s = 160$. In this case, the interleaver parameters given in Table 4.6 are used. In the second case, they are provided in Table 4.16, and bit puncturing is implemented to achieve $R = 1/2$. The corresponding FER curves are provided in Fig. 4.18.

When $K_s = 160$ GF(64) symbols, the NB-TC constructed from C_1 outperforms the 5G-LDPC code at both high and low error rates: this gain is of more than 0.7 dB at high error rates, while the error floor is lowered by more than two decades. When C_3 is considered, the NB-TC perform closely to the 5G-LDPC code at high error rates, while it shows a much lower error floor (in the order of 10^{-11} of FER).

On the other hand, for $K_s = 900$ GF(64) symbols, the NB-TC still outperforms the 5G-LDPC code by more than 0.3 dB at high error rates, while also showing lower error floor.

As a conclusion, the designed NB-TC can outperform the binary LDPC codes defined

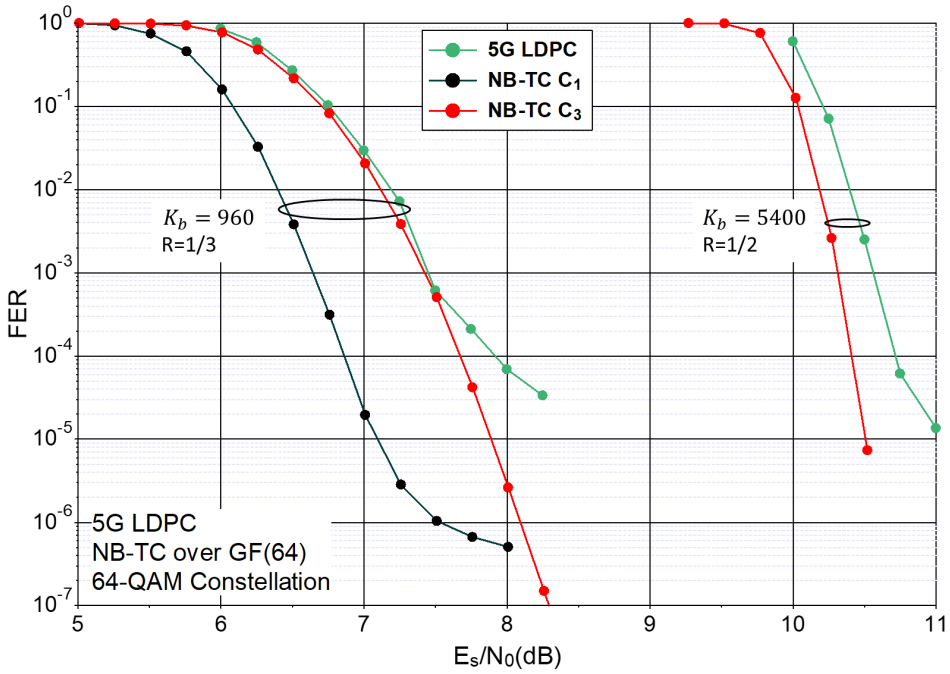

Figure 4.18: Frame error rate performance comparison of NB-TC over GF(64) and 5G-LDPC code, different frame sizes and coding rates, AWGN channel and 64-QAM constellation.

in the 5G standard, for different modulation orders, frame sizes, and coding rates. This better performance is made possible by the offered degrees of freedom in the NB structure, as well as the gap between the BICM and CM capacity curves.

4.6.3 Comparison with NB-LDPC codes

NB-LDPC codes are the first competitor for NB-TCs, since they adopt symbol encoding, with the same advantages in terms of capacity gains and freedom degrees. Different comparisons between these two types of codes are conducted for different frame sizes, constellation orders and coding rates.

The first set of comparisons consider a transmission using a 4-QAM constellation, and the performance curves for the NB-LDPC codes are taken from [96]. In Fig. 4.19, a first comparison is depicted for a very short frame size, where $K_s = 3$ GF(64) symbols and $R = 1/3$. For this comparison example, the NB-TC is shown to outperform the NB-LDPC code over GF(64) by 0.5 dB.

A second comparison is conducted for a 4-QAM constellation, with $K_s = 75$ GF(64) symbols and $R = 3/5$. The parameters of the ARP interleaver of the NB-TC are given in Table 4.17 and puncturing respects the mask of Table 4.18. Fig. 4.20 shows the frame error rate comparison between these two codes, where the NB-TC still slightly outperforms the NB-LDPC code.

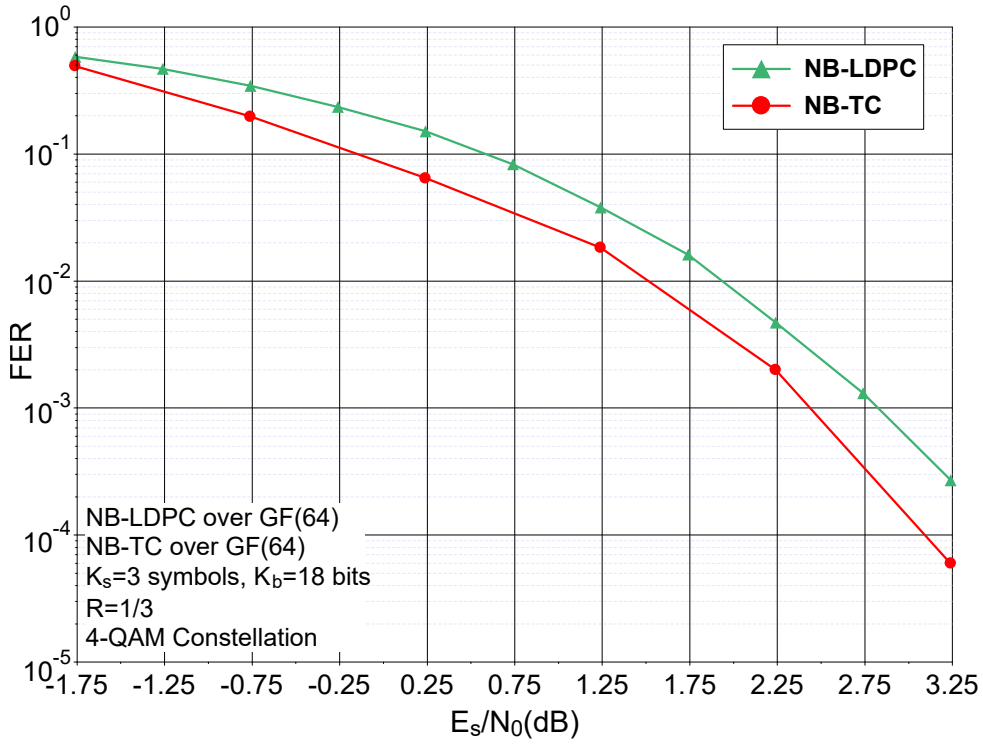

Figure 4.19: Frame error rate performance comparison of NB-TC and NB-LDPC code defined over GF(64), $K_s = 3$ GF(64) symbols and $R = 1/3$, AWGN channel and 4-QAM constellation.

P	Q	(S(0),...,S(Q-1))
68	5	(0,25,38,12,0)

Table 4.17: ARP interleaver parameters for $K_s = 75$ GF(64) symbols.

s	p_1	p_2
11111	00101	10010
11111	00000	01010
11111	10101	01000

Table 4.18: Puncturing mask for binary TC with $R = 3/5$, and $K_b = 450$ bits. The puncturing period is $M = 15$.

Moreover, a third comparison is performed for $K_s = 492$ GF(64) symbols and $R = 1/2$. Interleaver parameters of the NB-TC are provided in Table 4.19, and a regular symbol puncturing is applied as shown in Table 4.20. According to Fig. 4.21, the designed NB-TCs offer significantly better performance than the NB-LDPC code of [96] at high and low error rates for this simulation example.

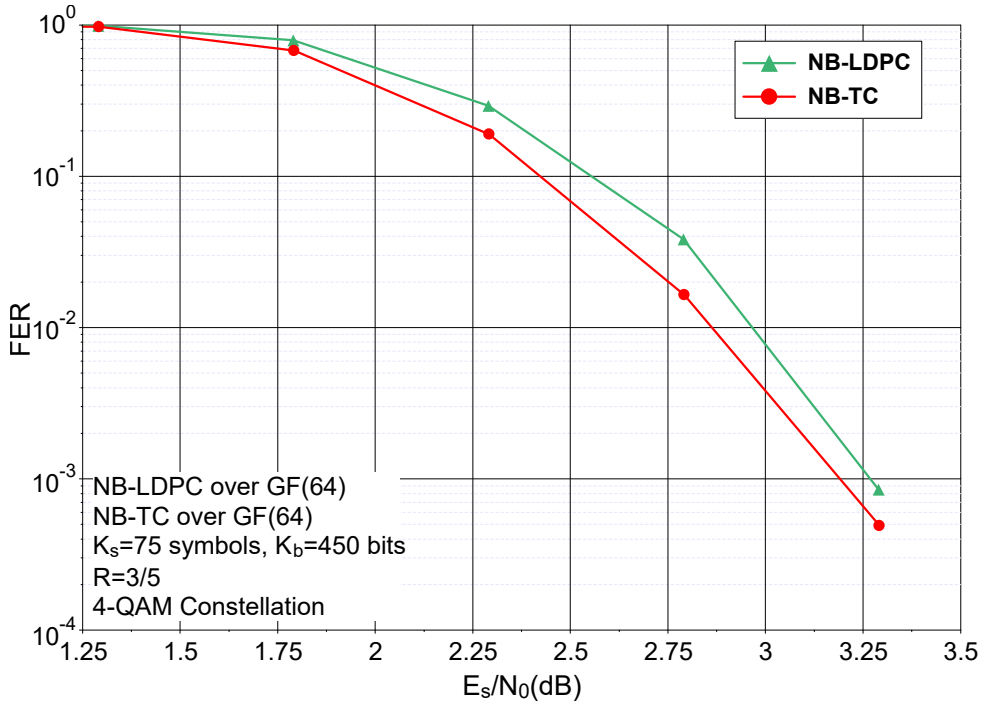

Figure 4.20: Frame error rate performance comparison of NB-TC and NB-LDPC code defined over GF(64), $K_s = 75$ GF(64) symbols and $R = 3/5$, AWGN channel and 4-QAM constellation.

P	Q	(S(0),...,S(Q-1))
23	4	(0,422,336,354)

Table 4.19: ARP interleaver parameters for $K_s = 492$ GF(64) symbols.

s	p_1	p_2
11	01	10

Table 4.20: Puncturing mask for binary TC with $R = 1/2$, and $K_s = 492$ GF(64) symbols. The puncturing period is $M = 2$.

A second set of comparisons is performed for a transmission using a 64-QAM constellation. First, our NB-TCs are compared with the NB-LDPC code of [97] in terms of Bit Error Rate (BER). Bit puncturing of Fig. 3.15 is considered for the NB-TC to achieve $R = 1/2$, and the parameters of the ARP interleaver are provided in Table 4.21. Fig. 4.22 shows the performance comparison between the NB-LDPC code and two NB-TCs constructed from different NB-CCs. In this example, NB-TCs are shown to outperform the NB-LDPC code by more than 0.25 dB.

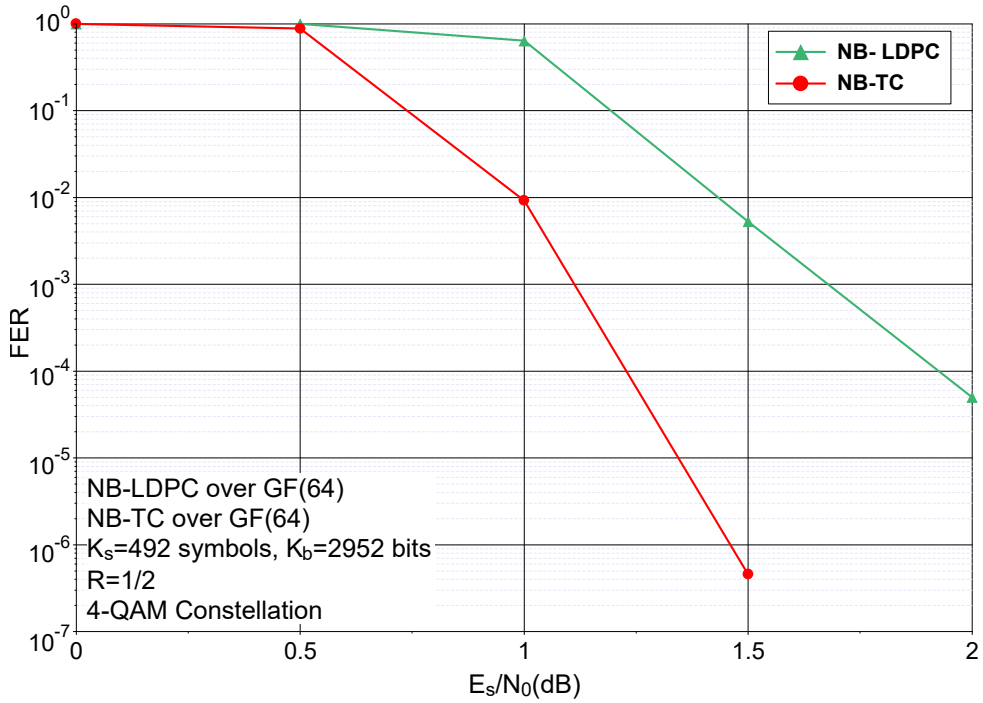

Figure 4.21: Frame error rate performance comparison of NB-TC and NB-LDPC code defined over GF(64), $K_s = 492$ GF(64) symbols and $R = 1/2$, AWGN channel and 4-QAM constellation.

P	Q	(S(0),...,S(Q-1))
31	4	(0,62,108,42)

Table 4.21: ARP interleaver parameters for K_s = 192 GF(64) symbols.

A second comparison is performed with the NB-LDPC code of [98], for $K_s = 256$ GF(64) symbols and $R = 2/3$. Table 4.22 shows the ARP interleaver parameters, and Fig. 4.23 describes the adopted bit puncturing to achieve coding rate $R = 2/3$, where s denotes the systematic value at different positions in the information frame, and p denotes the parities. This puncturing is based on enhancing the protection of the systematic symbols by using their corresponding bits at the 4 MSBs of each generated symbol, while one parity is considered out of each four, and its bits are sent at the LSBs of the generated symbols. Fig. 4.24 shows that a NB-TC using constituent code C_3 can outperform the NB-LDPC code of [98] by more than 0.3 dB.

P	Q	(S(0),...,S(Q-1))
11	4	(0,164,4,112)

Table 4.22: ARP interleaver parameters for K_s = 256 GF(64) symbols.

Whatever the source of the NB-LDPC code, we have been able to design a NB-TC that outperforms its LDPC counterpart. When considering short frame sizes, the performance gap is small but it tends to increase with the frame size.

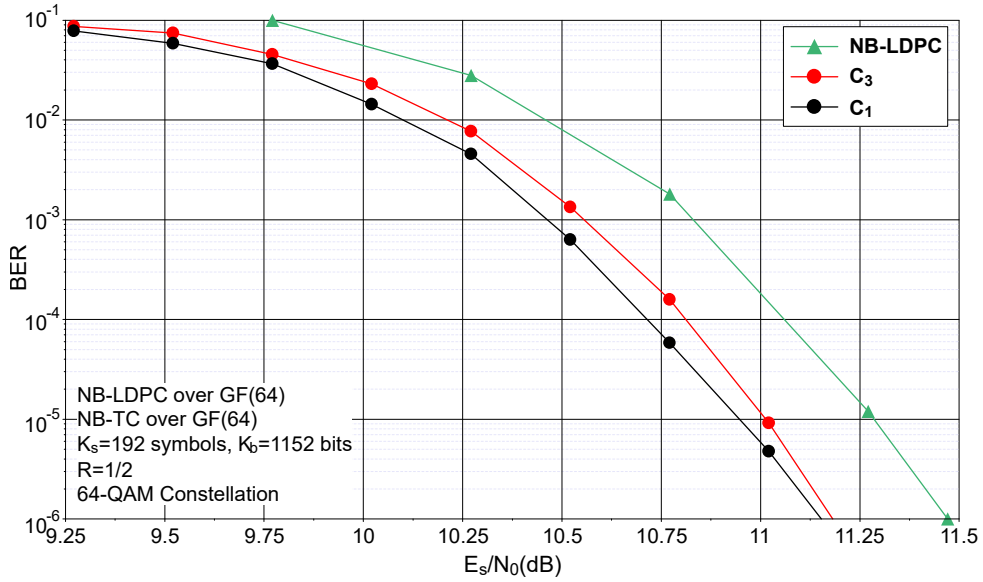

Figure 4.22: Bit error rate performance comparison of NB-TC and NB-LDPC code defined over GF(64), $K_s = 192$ GF(64) symbols and $R = 1/2$, AWGN channel and 64-QAM constellation.

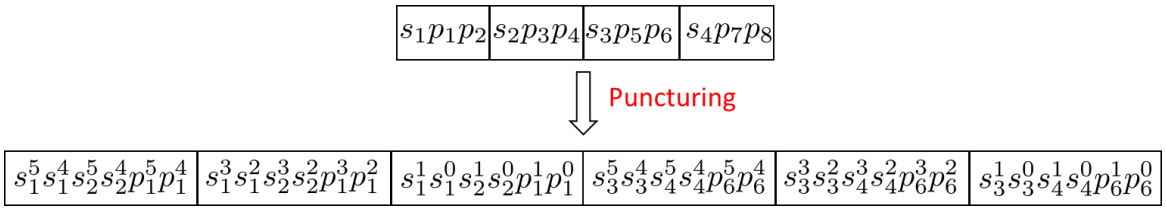

Figure 4.23: Bit puncturing structure for $K_s = 256$ GF(64) symbols and $R = 2/3$.

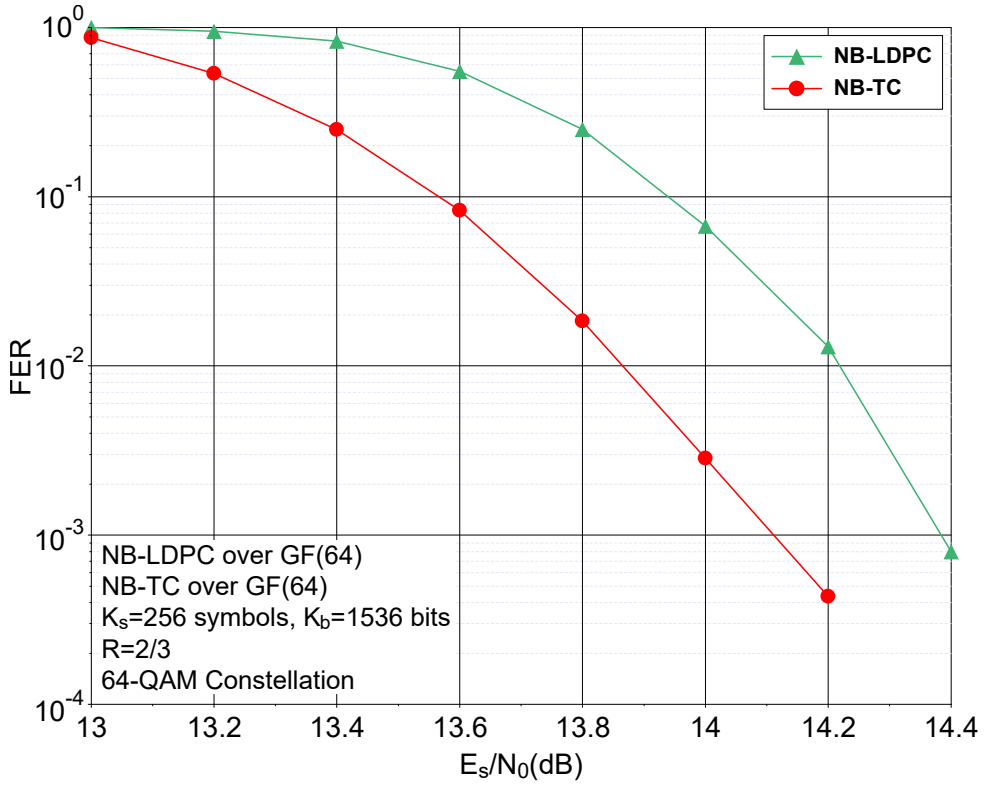

Figure 4.24: Frame error rate performance comparison of NB-TC and NB-LDPC code defined over GF(64), $K_s = 256$ GF(64) symbols and $R = 2/3$, AWGN channel and 64-QAM constellation.

4.6.4 Comparison with 5G-polar codes

Polar codes were proposed by Arıkan in [5], and were proven to achieve the theoretic capacity under different scenarios. These codes were adopted for the control channel in the 5G NR standard [93].

Different simulations were carried out to compare our NB-TCs with the 5G-polar codes, simulated from the Matlab code [99] using the Successive Cancellation List (SCL) decoder with different list sizes.

The first proposed set of error rate curves is obtained with a short frame size, here K_s = 10 GF(64) symbols equivalent to K_b = 60 bits, with two coding rates (R = 1/3 and R = 1/2). The interleaver of the NB-TC respects the parameters in Table 4.23, while the regular symbol puncturing in Table 4.20 is used to achieve R = 1/2. Fig. 4.25 shows the performance comparison, in terms of FER, between the NB-TC and the 5G-polar code simulated with an SCL decoder with $L = 8$ for the dashed line, and $L = 32$ for the continuous one. NB-TCs are shown to largely outperform polar codes decoded with $L = 8$, while a slight gain over polar codes with $L = 32$ is observed.

P	Q	(S(0),...,S(Q-1))
3	5	(0,4,8,2,6)

Table 4.23: ARP interleaver parameters for K_s = 10 GF(64) symbols.

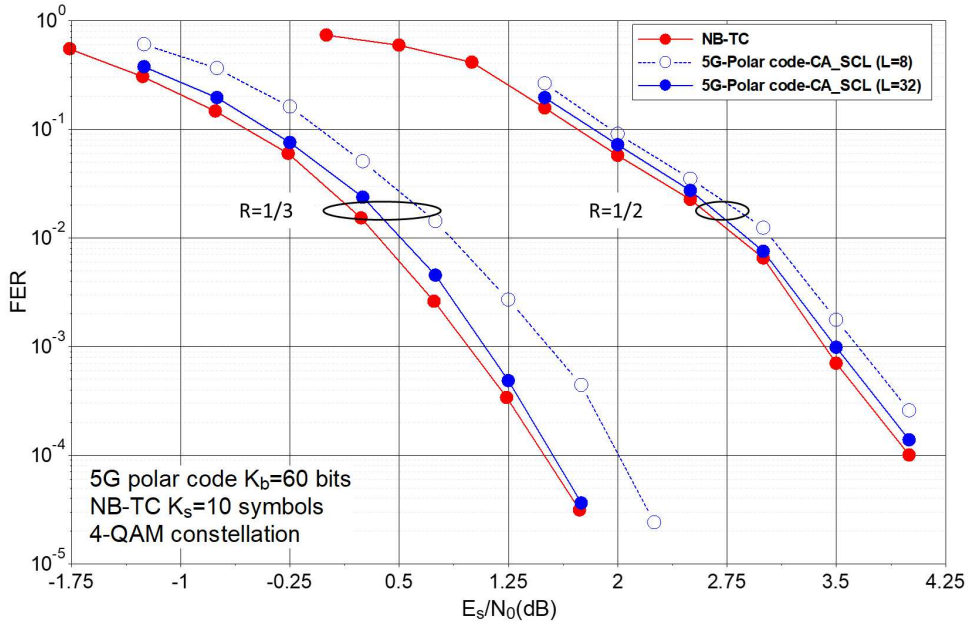

Figure 4.25: Frame error rate performance comparison of NB-TC defined over GF(64), K_s = 10 GF(64) symbols, and 5G-polar code, K_b = 60 bits, with different coding rates, AWGN channel and 4-QAM constellation.

The second simulation example is conducted for a moderate frame size, where K_s = 162 GF(64) symbols (equivalent to K_b = 972 bits) and a coding rate R = 9/10. The used interleaver and puncturing mask are shown in Tables 4.24 and 4.25. Fig. 4.26 shows that,

for longer data blocks, the NB-TC largely outperforms the polar code with $L = 8$ and $L = 32$ at a high coding rate. This gain is up to 1 dB.

P	Q	(S(0),...,S(Q-1))
49	9	(0,97,85,82,61,112,73,151,5)

Table 4.24: ARP interleaver parameters for K_s = 162 GF(64) symbols.

s	p_1	p_2
11110	10000	00001
1111	0000	0000

Table 4.25: Symbols puncturing mask for NB-TC with R = 9/10, and K_s = 162 GF(64) symbols. The puncturing period is $M = 9$.

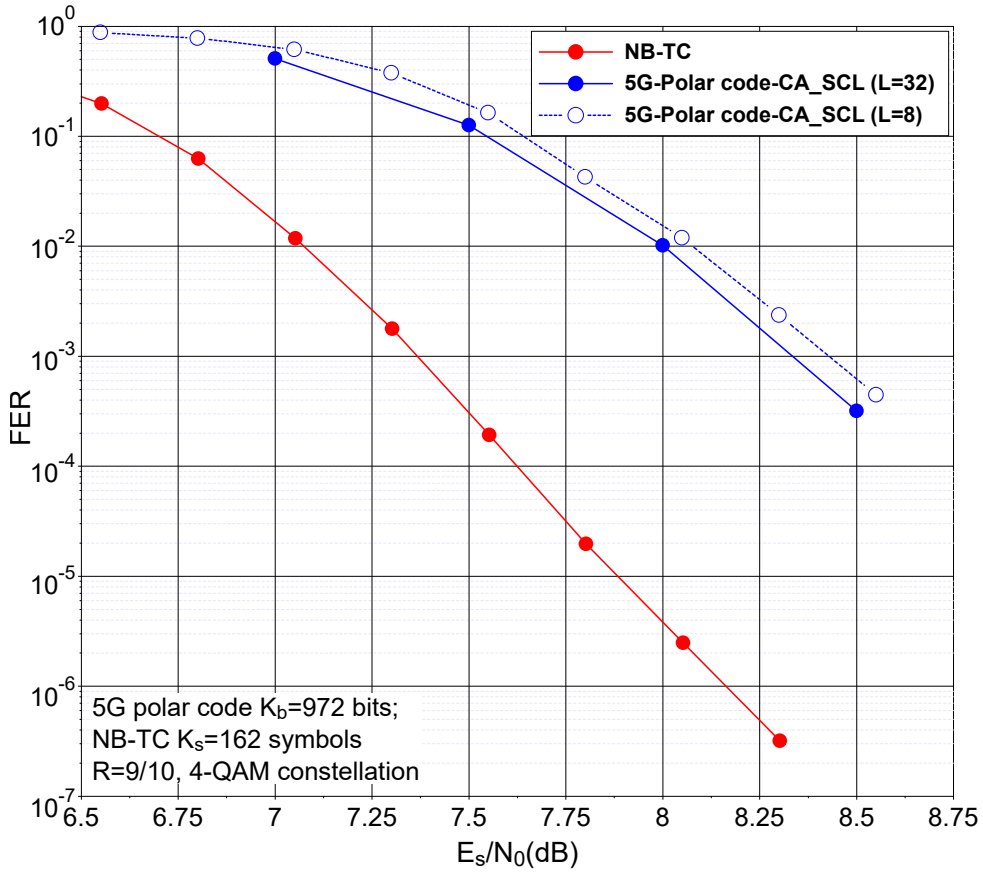

Figure 4.26: Frame error rate performance comparison of NB-TC defined over GF(64), $K_s = 162$ GF(64) symbols, and 5G-polar code, $K_b = 972$ bits with $R = 9/10$, AWGN channel and 4-QAM constellation.

A third comparison example is presented with K_s = 164 GF(64) symbols (equivalent to K_b = 984 bits) and two coding rates: R = 1/2 and R = 2/3. The interleaver parameters

for the NB-TC are shown in Table 4.26, while the regular symbol puncturing shown earlier is used to achieve $R = 1/2$, and the puncturing mask of Table 4.27 is proposed to achieve $R = 2/3$. Fig. 4.27 shows that the NB-TC still outperforms the polar codes for these medium size and coding rates by more than 0.5 dB.

P	Q	(S(0),...,S(Q-1))
147	4	(0,58,41,45)

Table 4.26: ARP interleaver parameters for $K_s = 164$ GF(64) symbols.

s	p_1	p_2
1111	0010	1000

Table 4.27: Symbol puncturing mask for NB-TC with $R = 2/3$, and $K_s = 164$ GF(64) symbols. The puncturing period is $M = 4$.

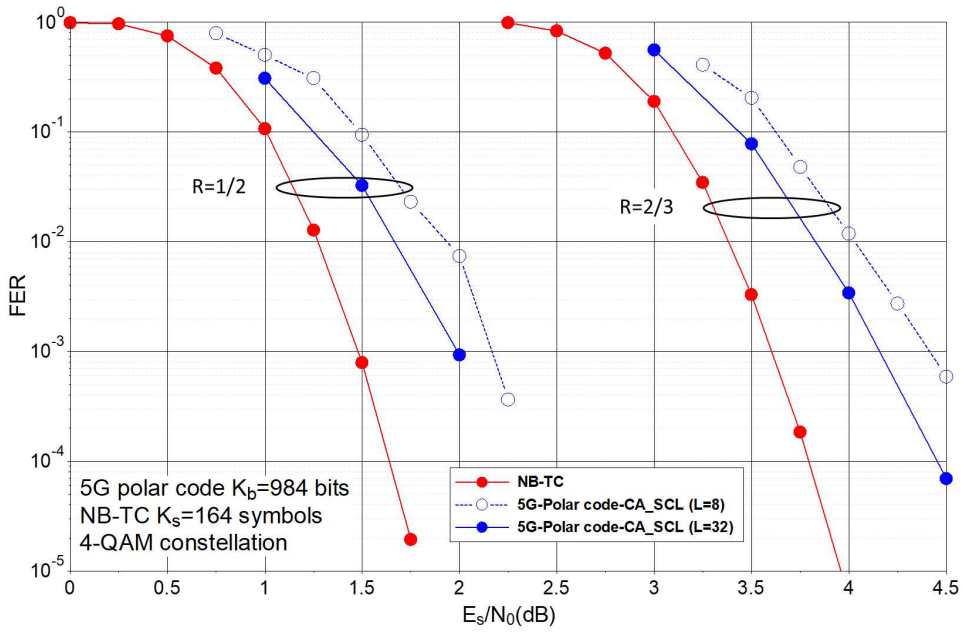

Figure 4.27: Frame error rate performance comparison of NB-TC defined over GF(64), $K_s = 164$ GF(64) symbols, and 5G-polar code, $K_b = 984$ bits with different coding rates, AWGN channel and 4-QAM constellation.

Even though the polar codes are known to achieve the capacity of a large class of channels, we showed in this section that NB-TCs can outperform these codes for different frame sizes and code rates in the AWGN channel.

4.6.5 Capacity achieving codes

Having obtained very low error floors with the proposed NB-TC structure, we further conduct a comparison of the achieved performance with respect to the theoretical capacity.

Code C_1, i.e. the NB-CC defined over GF(64) with the worst distance spectrum, is used in this section as NB-CC constituent code, since the resulting NB-TC shows a lower convergence threshold than with better NB-CCs. Using the proposed interleaving method, and the transformation block, acceptable asymptotic performance can be achieved by this code. Our goal here is to compare the actual error correction performance of the resulting NB-TC towards the achievable CM capacity.

To this end, simulations are conducted for a frame size K_s = 8000 GF(64) symbols corresponding to K_b = 48000 bits. An ARP interleaver is considered with the parameters defined in Table 4.28. The transformation described in Table 4.3 is also used. 16 iterations of the MAP decoding algorithm are performed, and the results are compared with the values of the BICM and CM capacities for a 64-QAM constellation and a coding rate $R = 1/3$. Fig. 4.28 shows the obtained results.

P	Q	(S(0),...,S(Q-1))
1507	16	(8,3443,7934,4420,7894,7903,511,7397, 2248,4672,1481,3464,1675,3116,7435,4735)

Table 4.28: ARP interleaver parameters for K_s = 8000 GF(64) symbols.

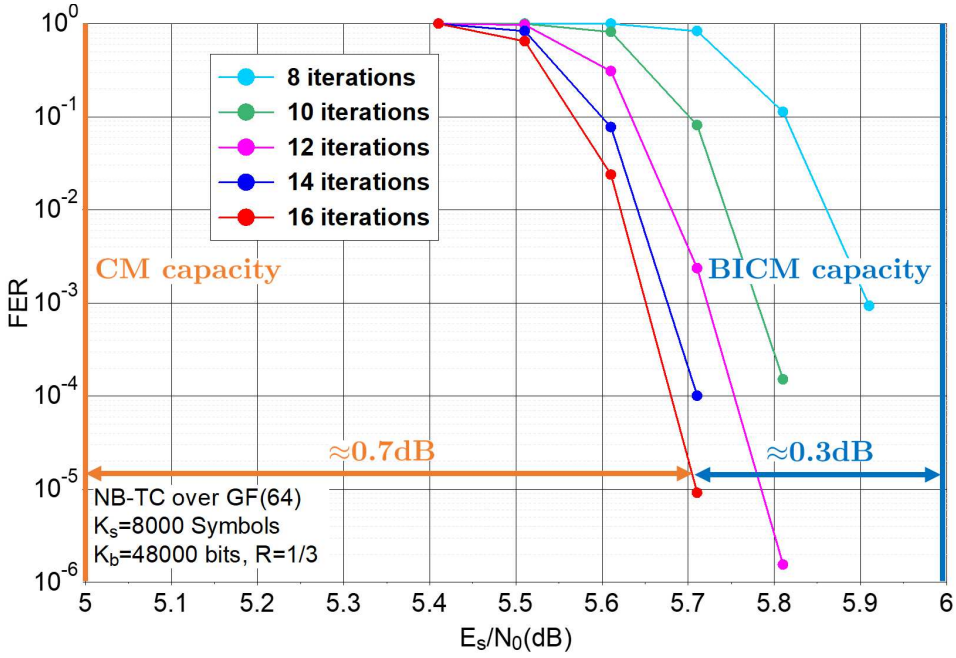

Figure 4.28: Frame error rate performance comparison of NB-TC defined over GF(64) and the BICM and CM capacities for K_s = 8000 GF(64) symbols and $R = 1/3$, AWGN channel and 64-QAM constellation.

We can observe that, at a FER of 10^{-5}, and after 16 decoding iterations, the NB-TC using C_1 as a constituent code performs beyond the BICM capacity: a gap of 0.3 dB can be observed. Moreover, the designed code can approach within 0.7 dB the CM capacity. This means that, for long data blocks, no optimization of binary codes can yield better performance than the designed code in the same transmission conditions.

4.7 Conclusion

In this chapter, a novel method to enhance the coding gain of NB-TCs was proposed. This method consists in introducing a new transformation block in the structure of these codes (symbol transformation before one of the component NB RSC encoders). This block changes the value of the symbols to be encoded by one of the NB constituent encoders. The aim is to increase the Euclidean distance between two constellation points, before or after transformation.

In this context, several design criteria were proposed for the design of the transformation, inspired from the structure of the used constellation. These criteria allowed us to define the best possible bijective transformations over GF(64) for 64-QAM and 4-QAM constellations.

Consequently, using these transformations, the observed performance gain was remarkable, showing a gap of more than a decade in the error floor region. This transformation block, requiring almost no additional transmit or receive complexity, can be used with any type of interleaver in order to considerably lower the error floor without any penalty on the excellent low SNR performance offered by the NB codes.

The obtained NB-TCs were then compared with the existing binary and NB codes from the literature, and we showed that our proposal outperforms the state-of-the-art codes under different scenarios.

Note that the addition of a transformation block into the structure of NB-TCs proposed in this chapter has been the subject of a patent filing. Also, a journal paper is being prepared targeting this technique.

Chapter 5

Design of low complexity decoders for non-binary turbo codes

In Chapters 2, 3 and 4, a new non-binary encoding structure is proposed. These new NB-TCs are designed over GF(q) and show attractive error correction capabilities, both in the waterfall end error floor regions. The designed codes are shown to outperform their binary counterparts either when mapped to their corresponding q-QAM constellation, or when mapped to lower order constellations such as QPSK. This gain is due to the capacity gains observed between coded modulations and bit-interleaved coded modulations, on the one hand, and to the additional degrees of freedom introduced into the non-binary structure, on the other hand.

The proposed NB-TC scheme is based on NB-CCs designed from an updated accumulator structure, which imposes codes with fully connected trellises. Respecting this structure, the decoding complexity grows with the order q of the finite field. The computational complexity of the symbol-based BCJR algorithm [72], when implemented in the logarithm domain [14], scales as $q^{\nu+1}$ in terms of addition and compare-select (ACS) operations, ν being the code memory. Also, the storage requirement for the extrinsic information varies linearly with q and the state metric memory is in the order of q^{ν} as will be shown in Section 5.3.1. For high values of q, the high complexity can limit the adoption of non-binary codes in communication standards, since hardware implementation becomes problematic. A few years ago, NB-LDPC codes faced the same type of complexity problems and several complexity reduction techniques were proposed in the literature.

The aim of this chapter is to propose a new decoding algorithm that reduces the decoding complexity while preserving the good performance of NB-TCs. Before describing the decoder, a state of the art in low-complexity decoding of NB-LDPC codes is presented in Section 5.1. This review helped us to develop decoding algorithms with similar properties for the case of NB-TCs. Next, the classical decoding algorithm for NB-TCs is presented in Section 5.2, alongside with the resulting decoding complexity. In Section 5.3, a new low-complexity decoding algorithm for NB-TCs is proposed. Firstly, a method to limit the storage requirements is presented in 5.3.1, and the decoding steps are explained in 5.3.2. A complexity analysis is then performed in Section 5.3.3, where the number of required operations is calculated for different decoding schemes. Simulation and decoding complexity results are provided in Section 5.4 for different types of NB-TCs and a comparison with the complexity of binary TCs decoders is carried out. Finally, Section 5.5

concludes the chapter.

5.1 State of the art in low-complexity decoders for NB-LDPC codes

Introduced by MacKay [19], NB-LDPC codes can be decoded following several approaches, in the same way as binary LDPC codes. The Belief Propagation (BP) algorithm for NB-LDPC codes was firstly introduced in [19] as a decoding template. Then, the Extended Min-Sum (EMS) algorithm was proposed in [100, 101], largely inspired from the min-sum decoding algorithm used for binary LDPC codes [102]. In [100, 103, 104], it is shown that EMS can be implemented with acceptable storage and computational complexities. The achieved complexity reduction is partly due to the truncation of the values of the exchanged messages, which results in limiting the complexity of the operations at the check nodes. Additional simplifications were introduced into the EMS decoding structure in order to reduce its complexity. The bubble check algorithm was proposed in [105, 106] as a tool allowing the simplification of the check node computations in the EMS algorithm. Inspired by the decoders designed for NB-LDPC codes, we were able to devise a simplified decoding algorithm for NB-TCs over GF(q). To this end, a literature review of the decoding algorithms for NB-LDPC codes is presented in this section.

5.1.1 The belief propagation decoding algorithm

As mentioned earlier in Chapter 1, a valid codeword c of a NB-LDPC code defined over GF(q) should satisfy $cH^T = 0$. In [19], it is shown that the decoding algorithm should find the most probable codeword c that verifies $cH^T = 0$. The one provided in [19] represents an extension of the BP decoding algorithm [107] to the non-binary domain.

The BP algorithm is based on the propagation of information iteratively between the variable nodes and the check nodes of a Tanner graph. When propagating through the graph, messages update the information on nodes until finding a valid codeword respecting all the parity check equations, or until achieving a maximum predefined number of iterations.

Considering the Tanner graph of Fig. 5.1, if C_i and V_j represent a check node and a variable node connected by an edge in the graph, the message output by C_i and sent to V_j through the edge is denoted by $C^O_{C_iV_j}$. Similarly, the messages entering a check node C_i, are represented by $C^I_{C_iV_j}$. At the variable node end, the messages are denoted by $V^I_{C_iV_j}$ and $V^O_{C_iV_j}$ for the input and the output messages respectively.

The BP algorithm can be divided into six main steps:

1. ***Initialization step:*** In this step, the information messages passing from the variable nodes V_j to the check nodes C_i, $V^O_{C_iV_j}$, are initialized with the *a priori* information received at the channel output. Let $P_{c_j}[a_p] = P(c_j = a_p \mid y)$ be the probability that codeword c at position j is equal to the symbol a_p, where $y = (y_0, \ldots, y_{N-1})$ is the received sequence of symbols, N is the codeword size, $a_p \in$ GF(q), and $p \in [0, q-1]$. The q values of $V^O_{C_iV_j}$ are set to $P_{c_j}[a_p]$ such that $V^O_{C_iV_j} = P_{c_j}[a_p]$ with C_i spanning the set of the check nodes connected to the variable node V_j.

2. ***Variable nodes update:*** Since each message $V^O_{C_iV_j}[a_p]$ passing from a variable node V_j to the connected d_v check nodes represents the probability that the codeword

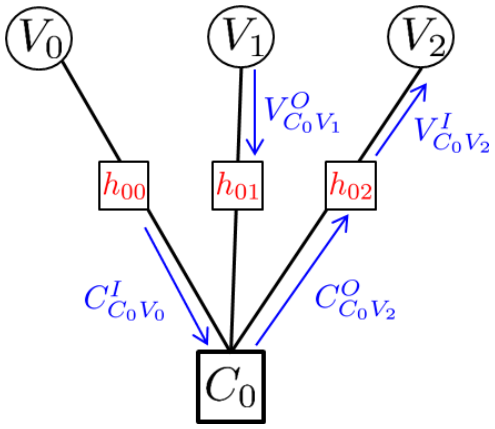

Figure 5.1: Example of a Tanner graph with one check node and three variable nodes, showing the process of message passing during BP algorithm.

symbol at position j is equal to a_p, they are updated through the following equation:

$$V^O_{C_iV_j}[a_p] = \mu P_{c_j}[a_p] \prod_{w=0,w\neq i}^{d_v-1} V^I_{C_wV_j}[a_p] \tag{5.1}$$

where μ represents a normalization factor ensuring $\sum_{a_p\in \mathrm{GF}(q)} V^O_{C_iV_j}[a_p]=1$.

3. ***Permutation nodes update:*** Since the coefficients in the parity check matrix of NB-LDPC codes are elements of GF(q), the VN messages are multiplied by non-binary coefficients h_{ij} before being transmitted to the connected CN. This multiplication can also be viewed as a permutation of the message value [19]. Consequently:

$$C^I_{C_iV_j}[a_p] = V^O_{C_iV_j}[a_p h_{ij}] \tag{5.2}$$

4. ***Check nodes update:*** The output messages $C^O_{C_iV_j}$ of every CN C_i are updated. Each output message represents the probability that the parity equation is verified depending on the input symbols a_p. $C^O_{C_iV_j}$ are calculated by:

$$C^O_{C_iV_j}[a_p] = \sum_{l\in\xi} \prod_{w=0,w\neq j}^{d_c-1} C^I_{C_iV_w}[a_l] \tag{5.3}$$

where V_j spans the d_c VNs connected to C_i, and ξ represents the set of GF(q) symbols whose sum equals a_p.

5. ***Permutation nodes update:*** Similarly to the third step, the message entering the VNs are calculated from the message leaving the CNs by simple permutations according to coefficients h_{ij}:

$$V^I_{C_iV_j}[a_p] = C^O_{C_iV_j}[a_p h_{ij}^{-1}] \tag{5.4}$$

6. ***Decision and stopping criterion:*** After each iteration, a decision on the codeword symbol is taken, according to:

$$\widehat{c_j} = \underset{a_p\in\mathrm{GF}(q)}{\mathrm{argmax}} \left[P_{c_j}[a_p] \prod_{w=0}^{d_v-1} V^I_{C_wV_j}[a_p] \right] \tag{5.5}$$

A test is performed to check whether $\widehat{c}H^T = 0$ or not. If $\widehat{c}H^T = 0$, the decoding is completed by adopting $c = \widehat{c}$; otherwise, a new iteration is performed unless the maximum number of iterations has been reached.

This algorithm is not suitable for hardware implementation, since it operates in the probability domain, requiring a high number of multiplications. To this end, it can be implemented in the logarithm domain, where the multiplications of probabilities are transformed into simple additions and comparisons of Log-Likelihood Ratios (LLR).

5.1.2 The extended min-sum algorithm

In order to reduce the complexity of the BP algorithm in the logarithmic domain, the Min-Sum (MS) algorithm was introduced in the case of binary LDPC codes [108, 109, 102, 110, 111]. Its extension for NB-LDPC codes was proposed in [100] and ameliorated in [101]. Truncating the messages at the input of the check node is proposed in [100], where only n_m reliabilities are used in the messages instead of q reliabilities, with $n_m < q$. Furthermore, in [101], all the messages in the Tanner graph are truncated to n_m values. The value of n_m should be carefully chosen to avoid large performance penalties.

Since the update of the check nodes represents the bottleneck of the decoding complexity, the EMS algorithm proposes a forward-backward computation method as a possible solution. The following steps are applied by the EMS algorithm:

1. ***Initialization step:*** Similarly to the initialization step of the BP algorithm, the messages passing from the VNs to the CNs are initialized with the channel LLR values. LLRs are calculated, for $i \in [0, N-1]$, as follows:

$$\mathrm{LLR}[a_p] = \log\left(\frac{P(c_j = \hat{a} \mid y_i)}{P(c_j = a_p \mid y_i)}\right) \tag{5.6}$$

   with $\hat{a} = \text{argmax}\ \{P(c_j = a \mid y_i), a \in \mathrm{GF}(q)\}$.
   Here, only the n_m lowest LLR values, corresponding to the symbols having the highest probabilities are taken into consideration.

2. ***Variable nodes update:*** In this step, the output messages from the VNs $V^O_{C_iV_j}$ are updated according to the connected CNs other than C_i, and to the channel LLR. Basically, this update is calculated using:

$$V^O_{C_iV_j}[a_p] = \mathrm{LLR}[a_p] + \sum_{w=0, w\neq i}^{d_v-1} V^I_{C_wV_j}[a_p] \tag{5.7}$$

   In order to reduce the calculation complexity, only the n_m most reliable LLRs are taken into consideration, resulting in the calculation of n_m values of $V^O_{C_iV_j}$, while the remaining values are truncated.

3. ***Permutation nodes update:*** Similarly to the BP algorithm,

$$C^I_{C_iV_j}[a_p] = V^O_{C_iV_j}[a_p h_{ij}] \tag{5.8}$$

4. ***Check nodes update:*** The forward-backward process can be divided into several elementary calculations cells denoted as the Elementary Check Nodes (ECN). Since every CN is connected to d_c VNs, $(d_c - 2)$ ECN are processed at each of the three algorithm layers: the forward, backward, and merge layers. At the entry of each ECN cell, two vectors U and V of size n_m are presented. Its output provides the n_m most reliable values resulting from the sum of U and V. This processing represents the complexity bottleneck of the decoding algorithm.

5. ***Permutation nodes update:*** Similarly to the third step,

$$V^I_{C_iV_j}[a_p] = C^O_{C_iV_j}[a_p h_{ij}^{-1}] \tag{5.9}$$

6. ***Stopping criterion:*** A decision is taken after each iteration, with a test if the identified $\widehat{c}$ forms a codeword or not:

$$\widehat{c}_j = \underset{a_p \in \mathrm{GF}(q)}{\operatorname{argmax}} \left[\mathrm{LLR}[a_p] + \sum_{w=0}^{d_v-1} V^I_{C_wV_j}[a_p] \right] \tag{5.10}$$

The main drawback of the EMS algorithm proposed in [100] resides in the check node computation, since a high number of additions and comparisons have to be performed at each ECN. Different variants are proposed in order to decrease this computational complexity in [101, 105, 106] and are presented next in Section 5.1.3.

5.1.3 The bubble check algorithm

Since the computations at the ECN of the EMS algorithm form the bottleneck of the complexity of this algorithm, a new simplified algorithm based on the well-known bubble sorting principle is presented in [105, 106], aiming at the reduction of the necessary number of addition and comparison operations. In the ECN computation step, two input vectors, U and V of size n_m are used, $U = (U_1, U_2, \cdots, U_{n_m})$ and $V = (V_1, V_2, \cdots, V_{n_m})$, representing the LLRs at the input of each ECN. The components of vectors U and V are sorted in increasing order. The aim is to calculate the output vector E which represents the output message, of size n_m, containing the sorted values of the set $\{U_i + V_j\}, (i,j) \in [1 \cdots n_m]^2$ with a limited number of operations.

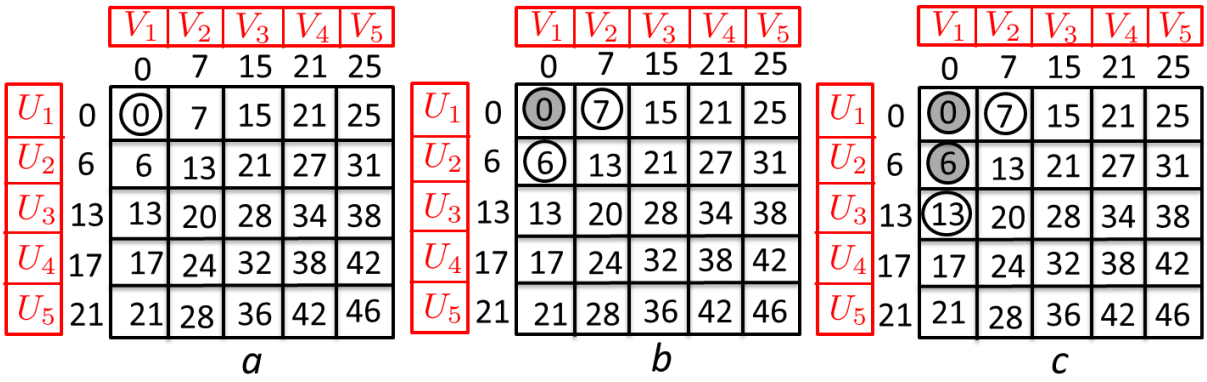

Figure 5.2: Example of *bubble check* processing.

An example is given in Fig. 5.2, where $n_m = 5$, $U = (0,6,13,17,21)$ and $V = (0,7,15,21,25)$. Each bubble $B_{i,j}$ located at row i and column j in the matrix, is calculated as the sum of the corresponding values of U_i and V_j: $B_{i,j} = U_i + V_j$. The algorithm begins by activating the bubble $B_{1,1}$, and the algorithm steps are the following:

1. ***Calculation of the next value of the output message:*** Choose the lowest value among the active bubbles. This value is chosen as the next value in the sorted output vector E and is removed from the list of active bubbles. In Fig. 5.2-a, the bubble $B_{1,1} = 0$ is chosen as the first value in E. In Fig. 5.2-b $B_{2,1} = 6$ is chosen as the second value of E, and $B_{1,2} = 7$ is chosen as the third value of E in Fig. 5.2-c.

2. ***Selection of the next candidate bubbles:*** After having chosen a bubble $B_{i,j}$ as the next output in vector E, two new bubbles can be considered for the comparison operation required to compute the next value in E: $B_{i+1,j}$ and $B_{i,j+1}$. However, if a previous candidate bubble is already active in the same row as $B_{i+1,j}$ or in the same column as $B_{i,j+1}$, the corresponding bubble cannot be considered in the candidate list. For instance, after having chosen $B_{1,1}$ in Fig. 5.2-a, $B_{2,1}$ and $B_{1,2}$ are the next candidates for the second iteration, as seen in Fig. 5.2-b. At the third iteration (Fig. 5.2-c), after having chosen $B_{2,1}$, the next potential candidates are $B_{3,1}$ and $B_{2,2}$. But since $B_{1,2}$ is already active in the same column as $B_{2,2}$, the candidate bubbles for the 3rd iteration are finally $B_{3,1}$ and $B_{1,2}$. This avoids the computation of $B_{2,2}$.

These two steps are iterated until the n_m values of the output message E have been calculated. This algorithm was shown to reduce the computational decoding complexity for the ECN computations from q^2 to the order of $n_m\sqrt{n_m}$ [105].

5.2 Min-Log-MAP decoding of NB-TCs and related complexity limitations

The well-known Max-log-MAP algorithm [14] tends to overestimate the extrinsic LLRs, especially during the first decoding iterations. Therefore, applying a scaling factor smaller than 1 before iterating yields a correction performance close to that of MAP decoding. To this end, the scaled Max-Log-MAP algorithm is introduced in [112], limiting the complexity of the conventional MAP decoder while approaching its performance. For the proposed NB-TCs, a variant of this algorithm extended to the symbol domain is adopted. The considered algorithm is based on the scaled Min-Log-MAP algorithm, which is equivalent to the scaled Max-Log MAP algorithm, except that we adopt the same LLR definition as in (5.6), which can also be expressed as the difference of two log-probabilities (LP):

$$\mathrm{LLR}(a) = \mathrm{LP}(a) - \min_{\alpha \in \mathrm{GF}(q)} \mathrm{LP}(\alpha) \tag{5.11}$$

where $\mathrm{LP}(a)$ represents the log-probability that the transmitted symbol c is a , knowing the received sequence y, $\mathrm{LP}(a) = -\log P(c = a \mid y)$.

With this definition, all the LLR values are greater than or equal to zero, and the LLR of the most probable symbol is zero. This LLR definition is considered in our study since it helps simplify the architecture of the decoder, as shown in [113].

5.2.1 Algorithm description

The recursive forward and backward processes of the Min-Log-MAP algorithm require the repeated computation of the minimum of cumulated terms (min-sum), for the derivation of the state metrics and of the extrinsic LLRs.

The forward state metric related to state $S_i = j$, $j \in \{0 \cdots q-1\}$, at trellis stage i is computed through the forward recursion as:

$$\alpha_i(j) = \min_{j' \in \{0 \cdots q-1\}} (\alpha_{i-1}(j') + \gamma_{s,i-1}(j', j) + \gamma_{p,i-1}(j', j)) \tag{5.12}$$

where $\alpha_{i-1}(j')$ is the forward state metric related to state $S_{i-1} = j'$ at trellis stage $i-1$, $\gamma_{s,i-1}(j', j)$ and $\gamma_{p,i-1}(j', j)$ are the systematic and parity transition metrics between states $S_{i-1} = j'$ and $S_i = j$, respectively. Similarly, the backward state metric related to state $S_i = j$ at trellis stage i is computed through the backward recursion as:

$$\beta_i(j) = \min_{j' \in \{0 \cdots q-1\}} (\beta_{i+1}(j') + \gamma_{s,i}(j, j') + \gamma_{p,i}(j, j')) \tag{5.13}$$

Additionally, the extrinsic LLR related to symbol $a \in \mathrm{GF}(q)$ at trellis stage i is computed using:

$$L_i^e(a) = \min_{(j,j') \in \{0 \cdots q-1\}^2 \,|\, s(j,j')=a} (\alpha_i(j) + \beta_{i+1}(j') + \gamma_{p,i}(j, j')) \tag{5.14}$$

where $s(j, j')$ is the systematic data symbol in $\mathrm{GF}(q)$ labeling transition (j, j').

5.2.2 Storage requirements of the Min-Log-MAP algorithm in the case of NB-TCs

First of all, q LLRs computed by the demapper have to be memorized at the input of the decoder for each received systematic and parity symbol, i.e. a total amount of qK/R input LLRs have to be stored, if K is the information block length and R the coding rate. A memory is also required to store the qK extrinsic LLRs exchanged during the iterative decoding process. Finally, each Min-Log-MAP decoder has also to store a certain amount of forward or backward metrics during the decoding process. In the case of a conventional window decoding process, qW forward or backward metrics have to be memorized, if W is the window length.

The required storage capacity grows linearly with the order q of the finite field. When targeting high order fields, storage requirements become the first obstacle for designing and implementing such decoders.

5.2.3 Computational complexity of the decoding algorithm in the case of NB-TCs

The calculation of a state metric α or β requires $q-1$ compare and select operations, which results in $q(q-1)$ compare and select operations for the calculation of the q state metrics of a trellis stage. Moreover, the same number of operations is required for the calculation of the extrinsic values of the q possible symbol values in $\mathrm{GF}(q)$, as shown in Eq. (5.14). In conclusion, the decoding complexity increases quadratically with the order, q, of the finite field. For high values of q, such as 64, 256 or larger, the decoding complexity becomes prohibitive and cannot be efficiently implemented in hardware without simplification. In

order to reduce this complexity, the required number of operations should be limited. This limitation should only entail an acceptable loss in error correction performance. Therefore, in the following, a new low-complexity decoding algorithm for NB-TCs is proposed, aiming at limiting the number of add-compare operations at each of the state metrics and extrinsic calculations. The proposal is inspired by the bubble check algorithm proposed in [105] and introduced in Section 5.1.3 to simplify the elementary check node processing of the extended min-sum algorithm for NB-LDPC codes.

5.3 Proposed low-complexity decoding algorithm for non-binary turbo codes

Two essential complexity limitations are identified in the decoding process of NB-TCs: storage requirements and computational complexity. Targeting an implementable version of the decoding algorithm, these two complexity types are addressed in the sequel.

5.3.1 Reduction of the storage requirements

Without modifications, high storage requirements are imposed by the proposed structure of the code over GF(q). We showed that storage complexity increases linearly with the order of the defined finite field. In order to propose an implementable structure of NB turbo decoders, one should reduce this complexity by limiting the number of quantization bits used to represent the data processed by the decoder, and by proposing a vector truncation in order to limit the number of memorized information from each vector. These propositions should be studied such that no penalizing performance losses are observed.

5.3.1.1 Fixed-point representation of decoder data

At the receiver end, each symbol is represented by q channel LLRs computed by the demapper. Let us consider that the signal is transmitted using a standard two-dimensional modulation and let I_r and Q_r be the in-phase and quadrature values of the received signal r. When a q-QAM constellation is considered for the transmission of symbols encoded by a NB-TC over GF(q), each symbol in the codeword is directly mapped to its corresponding constellation point. In other words, a symbol from a codeword having a value $a \in \mathrm{GF}(q)$ is mapped to (I_a, Q_a) in the q-QAM constellation. Assuming a transmission over an AWGN channel, the LLRs computed by the demapper at the receiving side can be expressed as follows, according to Eq. (5.11):

$$\mathrm{LLR}(a) = \mathrm{LP}(a) - \min_{\alpha \in \mathrm{GF}(q)} \mathrm{LP}(\alpha) \tag{5.15}$$

with

$$\mathrm{LP}(a) = \frac{(I_r - I_a)^2 + (Q_r - Q_a)^2}{2\sigma^2},\ a \in \mathrm{GF}(q) \tag{5.16}$$

where σ represents the standard deviation of the AWGN.

The LLR values resulting from Eq. (5.15) are real numbers. The hardware implementation of floating point operations being very demanding in terms of silicon area and power consumption, practical hardware implementations of decoders resort to fixed-point

representations and processing of the LLRs. Assuming that we adopt a conventional uniform quantization, the number of bits required for the quantization of the LLRs depends on the range of the LLRs values to be represented and on the required precision on the quantized values.

According to Eq. (5.15), the minimum LLR value is zero. For constellation points located far from the received signal, LLRs can take very large values. However, it is unlikely that very high LLR values participate in low ED paths. Therefore, large LLR values can be truncated without any error correction performance loss. In practice, we propose to apply the well-known 3σ rule of thumb and to truncate the LLR values for all the constellation points located at a distance greater than 3σ from the received signal, since there is 99.7% chance that the actual transmitted constellation signal is in this region. The highest LLR value can therefore be computed from Eq. (5.15) with $I_r - I_a = Q_r - Q_a = 3\sigma$, which gives $\mathrm{LLR}_{\max} = 9 - \min_{\alpha \in \mathrm{GF}(q)} \mathrm{LP}(\alpha)$. Whatever the value of the minimum LLR is, $\mathrm{LLR}_{\max} \leq 9$ and the range of the LLR values is taken equal to $[0,9]$.

For a FEC decoder, the choice of the quantization step size has a direct impact on the error correction performance at the decoder output. More specifically, it should be chosen in such a way that the minimum discrimination value between two constellation points guarantees a negligible loss in performance at the decoder output.

Let us consider a normalized constellation with d being the minimum distance between two points ($d = \sqrt{\frac{6}{q-1}}$ for a q-QAM constellation) and let us denote by x the minimum discrimination value for the chosen quantization. As shown in Fig. 5.3, if the received signal falls inside the circle of radius xd, the distance to the closest constellation point is taken equal to zero. Therefore the lowest non-zero LP value is equal to:

$$\mathrm{LP}_{\min} = \frac{(xd)^2}{2\sigma^2} \tag{5.17}$$

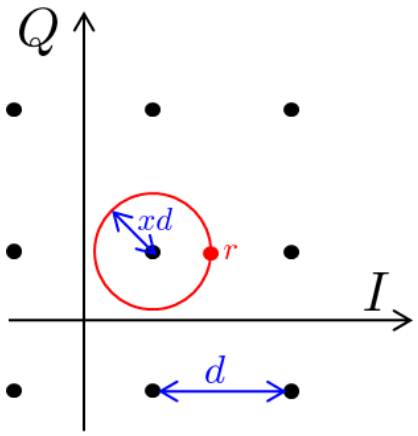

Figure 5.3: Illustration of the minimum discrimination radius.

To obtain a numerical value for this expression, one has to choose some values for σ and x. Actually, the value of σ is calculated from the capacity curve of the corresponding coded modulation (see Section 1.2.3.2), since the capacity sets a lower bound on the signal-to-noise ratio and consequently an upper bound on the noise variance to enable error correction. Then, the value of x is chosen to guarantee a negligible loss in performance at the decoder output (typically below 0.05 dB).

Taking the example of a 64-QAM constellation and coding rate $R = 1/3$, Eq. (5.17) is calculated with $(\frac{1}{2\sigma^2})_{\text{dB}} = \text{SNR}_{CM} = 5\text{dB}$ and simulations showed that choosing $x = 0.4$ entails a performance loss below 0.05 dB. Therefore, $\text{LP}_{\min} = \frac{(0.4)^2 \times 6/63}{1/\sqrt{10}} \approx 4.83 \times 10^{-2}$. The minimum number of values needed for the LLR representation is therefore $\frac{\text{LLR}_{\max}}{\text{LP}_{\min}} \approx 187$. Consequently, 8 bits should be used for the quantization of the LLR values, with the highest values truncated to integer 255.

Coming back to the description of the Min-Log-MAP algorithm (see Eq. (5.12) and (5.13)), the representation of the sum of the systematic and parity transition metrics, each quantized on 8 bits, requires 9 bits. These sums are then accumulated to compute the forward and backward state metrics. At each trellis step, the state metrics are normalized and truncated. In practice, it appears that truncating them to 511 (9 bits) does not entail any additional performance loss at the decoder output. This is due to the fact that amongst the 64 different metric values only a few of them have significant values, thus allowing harsh truncations. The same holds for the computation of the extrinsic LLRs from Eq. (5.14). To summarize, input LLRs are represented using 8 bits, internal state metrics and extrinsic LLRs are represented using 9 bits.

5.3.1.2 Vector truncation of decoder parameters

In the Min-Log-MAP decoding algorithm, four main data vectors are used: γ_s, γ_p, α and β. All these vectors consist of q components at each trellis stage.

γ_s and γ_p represent the transition metrics of the channel when systematic and parity symbols are transmitted using a q-ary modulation. α and β represent the forward and backward state metrics respectively, at trellis stage i, i spanning the q trellis states.

Since all these vectors are composed of q values each, their truncation to a limited number of representative values represents an efficient way to reduce the memory requirements. First of all, the vector components are sorted in increasing order, which corresponds to the increasing reliability order according to the adopted LLR definition. Then, only the first n_m lower values, with $n_m < q$, are selected and stored. When processing the decoding algorithm, the remaining $q - n_m$ values are given a low reliability value η, which is taken equal to the n_m^{th}, i. e. the least reliable, stored value.

The value of n_m is chosen based on performance simulation, so as to find a good compromise between the complexity reduction and the minimization of the resulting performance loss.

5.3.2 Simplified min-sum processing for NB-TCs

5.3.2.1 Simplified integer array sorting by addresses

The proposed decoding algorithm for NB-TCs, described in Section 5.3.2.2, is based on a modified bubble check algorithm and uses sorted arrays. Therefore, this section describes the design of a low-complexity sorting method.

The proposed sorter takes advantage of the fact that the LLR values and the decoder metrics are calculated in the integer domain thanks to the quantization proposed in Section 5.3.1.1, and that the LLR values are positive.

Let A be the array to sort in ascending order, and A_i the element of A at address i. The proposed sorter consists in storing the addresses i in a second array B, where $B_{A_i} = i$ as described in Fig. 5.4.

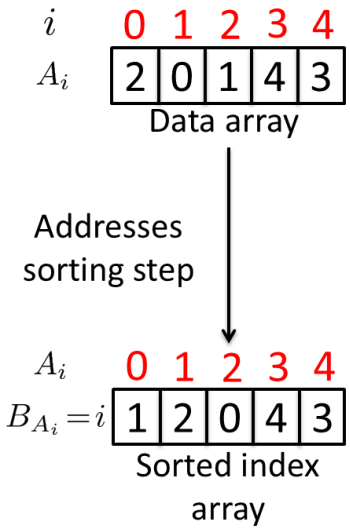

Figure 5.4: Integer array sorting by addresses.

As shown in Fig. 5.4, an integer array A can have its addresses in a sorted form in a second array B. It represents a kind of sorting by content value. Since $A_{B_i} \leq A_{B_{i+1}}$, B can be read from left to right in order to find the addresses in the sorted form for the data array A.

5.3.2.2 Reduced complexity decoding of NB-TCs

The adopted Min-Log-MAP decoding algorithm, as described in Section 5.2, requires a high computational complexity at each trellis stage, where $q(q-1)$ additions and comparisons are performed during the two recursive processes and the extrinsic calculation. For each of these processes, $(q-1)$ addition and comparison operations are needed for each of the q values in the alphabet GF(q).

However, the bubble check algorithm [105] as described in Section 5.1.3 cannot be directly applied for the min-sum processing of metrics and LLRs in the Min-Log-MAP algorithm, due to several reasons:

- min-sum processing does not require any sorting of the LLRs or cumulated metrics: only their minimum value has to be found;
- the computation of each cumulated term in Eq. (5.12), (5.13) and (5.14) involves the sum of three terms instead of two: sorting the cumulated values according to the standard bubble check algorithm would require running it twice;
- all the combinations of values are not possible in each sum: in other terms, some cells in the sorting table are empty.

Nevertheless, the proposed modified bubble check algorithm can still be used to simplify the decoding of NB-TCs. Considering the calculation of the forward metric $\alpha_i(j)$, q

transitions merge from trellis stage $i-1$ to the state j at trellis stage i. These q transitions define the connections between α_{i-1}, $\gamma_{s,i-1}$ and $\gamma_{p,i-1}$ that are used in the calculation of $\alpha_i(j)$ according to Eq. (5.12). The sorting table of the proposed algorithm contains the q values of the different terms $B_{j',j} = \alpha_{i-1}(j') + \gamma_{s,i-1}(j',j) + \gamma_{p,i-1}(j',j)$, called bubbles, by analogy with [105]. Since the computation of each cumulated term in Eq. (5.12), (5.13) and (5.14) involves the addition of three metric terms instead of two, using the standard bubble check algorithm of [105] would require running it twice. In order to keep the decoding complexity low, we propose a modified version of this algorithm that still uses a two-dimensional sorting table and that handles the third metric separately. A question that arises is the choice of the metric to be handled separately in the process. Although any metric could play this role, in practice it is better to keep in the table the metrics that are refined during the iterative decoding process thanks to the incoming extrinsic information, i.e. the state metric and the systematic transition metric: this speeds up the computation over the iterations. To this end, the rows and columns of the table are arranged according to the values of the forward metrics $\alpha_{i-1}(j')$ and the values of the systematic transition metrics $\gamma_{s,i-1}(j',j)$, both sorted in increasing order: $\{\alpha_{k_\alpha}\}, k_\alpha = 1 \cdots q$ and $\{\gamma_{s,k_s}\}, k_s = 1 \cdots q$. Each bubble value $B_{j',j}$ is placed in the table at the intersection of the corresponding $\alpha_{i-1}(j')$ and $\gamma_{s,i-1}(j',j)$ values. The parity transition metrics $\gamma_{p,i-1}(j',j)$ are sorted separately in increasing order following index $k_p = 1 \cdots q$.

An example of a sorting table is depicted in Fig. 5.5, for a NB-TC over GF(8). This table does not take into account the vector truncation described in Section 5.3.1.2. Its rows are indexed by the different values of the systematic transition metric vector $\gamma_{s,i-1}$, where $\gamma_s^k = \gamma_{s,i-1}(j',j)$, sorted in increasing order $\gamma_s^k \leq \gamma_s^{k+1}$, $k = 1 \cdots q$, and its columns are indexed by the different values of the forward state metric vector α_{i-1} with $\alpha^{k'} = \alpha_{i-1}(j')$, sorted in increasing order $\alpha^{k'} \leq \alpha^{k'+1}$, $k' = 1 \cdots q$. The table contains q^2 bubble values $B_{j',j}$ computed as $B_{j',j} = \alpha_{i-1}(j') + \gamma_{s,i-1}(j',j) + \gamma_{p,i-1}(j',j)$, where indexes j and j' refer to the state number in the trellis, not to the position in the table, which is given by k and k'. Since the structure of the considered NB-CC is fully connected, each state metric $\alpha^{k'}$ is connected to all the transition metrics γ_s^k ($k = 1 \cdots q$) and each term $B_{j',j}$, $(j',j) \in \{0,...,q-1\}^2$ corresponds to an actual transition in the trellis.

For each considered state $j = 0 \cdots q-1$, some bubbles from the table of Fig. 5.5 are considered, and others are not. This is due to the fact that, for each state j' at trellis stage $i-1$, only one systematic value $\gamma_{s,i-1}(j',j)$ can link the state j' to j in the trellis. To this end, when the calculation of $\alpha_i(j)$ is performed, only one bubble at each row and column can be considered.

5.3.2.2.1 Preliminary complexity reduction step

Before getting into the computation of each $\alpha_i(j)$, a first complexity reduction step is proposed. A radius R is identified by different simulation tests, which sets the boundaries of the table region corresponding to the R^2 lowest, i.e. most reliable, values of $\alpha_{i-1}(j') + \gamma_{s,i-1}(j',j)$. Before starting the bubble check procedure to compute $\alpha_i(j)$, the algorithm checks whether at least one bubble $B_{j',j}$ exists in this defined region. When the radius-R region does not contain any bubble corresponding to the arrival state j, it is highly unlikely that j has been visited by the encoder, since there exist no connection between the most reliable state metrics and the systematic transitions that can result in visiting state j. Therefore, the corresponding $\alpha_i(j)$ value is set to a predefined high value, depending on the chosen number of quantization bits. If a too small value is chosen for R, some states

	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8
γ_s^1	$B_{2,3}$	$B_{0,4}$	$B_{1,5}$	$B_{7,0}$	$B_{6,7}$	$B_{5,2}$	$B_{3,1}$	$B_{4,6}$
γ_s^2	$B_{2,6}$	$B_{0,0}$	$B_{1,3}$	$B_{7,2}$	$B_{6,1}$	$B_{5,7}$	$B_{3,4}$	$B_{4,5}$
γ_s^3	$B_{2,0}$	$B_{0,1}$	$B_{1,6}$	$B_{7,4}$	$B_{6,2}$	$B_{5,5}$	$B_{3,3}$	$B_{4,7}$
γ_s^4	$B_{2,5}$	$B_{0,7}$	$B_{1,1}$	$B_{7,6}$	$B_{6,4}$	$B_{5,0}$	$B_{3,2}$	$B_{4,3}$
γ_s^5	$B_{2,4}$	$B_{0,5}$	$B_{1,2}$	$B_{7,3}$	$B_{6,0}$	$B_{5,6}$	$B_{3,7}$	$B_{4,1}$
γ_s^6	$B_{2,1}$	$B_{0,2}$	$B_{1,0}$	$B_{7,7}$	$B_{6,5}$	$B_{5,3}$	$B_{3,6}$	$B_{4,4}$
γ_s^7	$B_{2,7}$	$B_{0,3}$	$B_{1,4}$	$B_{7,5}$	$B_{6,6}$	$B_{5,1}$	$B_{3,0}$	$B_{4,2}$
γ_s^8	$B_{2,2}$	$B_{0,6}$	$B_{1,7}$	$B_{7,1}$	$B_{6,3}$	$B_{5,4}$	$B_{3,5}$	$B_{4,0}$

Figure 5.5: Example of sorting table used for the computation of α_i using Eq. (5.12) for $q = 8$. All $B_{j',j}$ values, $j, j' = 0 \cdots q-1$ are placed in different rows and columns.

j with high probabilities won't be considered in the computation process and this will result in a non-negligible correction performance loss. On the other hand, high values of R result in low complexity reductions. The choice of R is the result of a compromise between the offered complexity reduction and the loss in the error correction capability of the NB-TC. To this end, for each Galois field, different values of R are tested, and the choice is based on the accepted performance loss. Going back to the example over GF(8) shown in Fig. 5.5, the proposed preliminary complexity reduction step is applied with $R = 3$. The resulting zone is depicted in blue in Fig. 5.6. In this example, bubbles corresponding to states $j = 2$ and $j = 7$ are shown to be out of the corresponding zone and then, the proposed algorithm automatically considers these states as not probable. Let η denote the highest quantized LLR value, the algorithm sets $\alpha_i(2) = \alpha_i(7) = \eta$, without any further calculations for these two states. On the contrary, at least one corresponding bubble $B_{j',j}$, for states $j \in \{0,1,3,4,5,6\}$ exists in the zone defined by the radius R in Fig. 5.6, and states $0,1,3,4,5$ and 6 are considered as probable ones. Therefore, the algorithm continues to calculate their corresponding $\alpha_i(j)$ according to the flowchart of Fig. 5.8.

5.3.2.2.2 Bubble processing

Following the truncation proposed in Section 5.3.1.2, not all of the values of GF(q) are taken into consideration for the state metric and transition vectors. Instead, a limited number n_m of values is chosen to be saved from each vector.

For each state $j \in [0, q-1]$, if the radius-R region contains one bubble or more, the computation process can be launched according to the flowchart of Fig. 5.8. The proposed algorithm processes the bubbles alternately vertically and horizontally (or *vice versa*) and updates two upper bounds, $k_{\alpha,\max}$ and $k_{s,\max}$. The bounds defined by $k_{\alpha,\max}$ and $k_{s,\max}$ are introduced to further dynamically limit the computation zone of the algorithm, where three indexes k_α, k_s and k_p point to the most reliable values, which have not been previously visited by the algorithm, in the corresponding vectors: α, γ_s and γ_p respectively. k_α varies in $[1, k_{\alpha,\max}]$, similarly k_s varies in $[1, k_{s,\max}]$, and finally since

	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8
γ_s^1	$B_{2,3}$	$B_{0,4}$	$B_{1,5}$	$B_{7,0}$	$B_{6,7}$	$B_{5,2}$	$B_{3,1}$	$B_{4,6}$
γ_s^2	$B_{2,6}$	$B_{0,0}$	$B_{1,3}$	$B_{7,2}$	$B_{6,1}$	$B_{5,7}$	$B_{3,4}$	$B_{4,5}$
γ_s^3	$B_{2,0}$	$B_{0,1}$	$B_{1,6}$	$B_{7,4}$	$B_{6,2}$	$B_{5,5}$	$B_{3,3}$	$B_{4,7}$
γ_s^4	$B_{2,5}$	$B_{0,7}$	$B_{1,1}$	$B_{7,6}$	$B_{6,4}$	$B_{5,0}$	$B_{3,2}$	$B_{4,3}$
γ_s^5	$B_{2,4}$	$B_{0,5}$	$B_{1,2}$	$B_{7,3}$	$B_{6,0}$	$B_{5,6}$	$B_{3,7}$	$B_{4,1}$
γ_s^6	$B_{2,1}$	$B_{0,2}$	$B_{1,0}$	$B_{7,7}$	$B_{6,5}$	$B_{5,3}$	$B_{3,6}$	$B_{4,4}$
γ_s^7	$B_{2,7}$	$B_{0,3}$	$B_{1,4}$	$B_{7,5}$	$B_{6,6}$	$B_{5,1}$	$B_{3,0}$	$B_{4,2}$
γ_s^8	$B_{2,2}$	$B_{0,6}$	$B_{1,7}$	$B_{7,1}$	$B_{6,3}$	$B_{5,4}$	$B_{3,5}$	$B_{4,0}$

Figure 5.6: Example of sorting table used for the computation of α_i using Eq. (5.12) for $q = 8$, with the blue zone corresponding to the preliminary complexity reduction step with $R = 3$.

the parity transition metric vector is handled separately in the process, $k_p \in [1, n_m]$. At the beginning of the algorithm, $k_{\alpha,\max} = k_{s,\max} = n_m$. These values are updated throughout the algorithm in order to decrease the calculation zone depending on the achieved reliability of the calculated value. In the example of the calculation of $\alpha_i(0)$, corresponding bubbles ($B_{j',0}$) are considered as shown in Fig. 5.7, while other bubbles ($B_{j',j''}$, with $j'' \neq 0$) are not considered in this step. State $j = 0$ is considered probable since the R-zone contains two bubbles for this state. Furthermore, $n_m = 5$ and $k_{\alpha,\max} = k_{s,\max} = n_m$. The gray zone depicts the restricted zone for the calculation, consequently bubbles $B_{1,0}$, $B_{3,0}$, $B_{4,0}$ and $B_{5,0}$ are not considered in the calculation process. More bubbles can be discarded by the algorithm at a later stage depending on the update of $k_{\alpha,\max}$ and $k_{s,\max}$ throughout the algorithm.

The steps of this algorithm are divided into horizontal and vertical processing. When a bubble is identified in the horizontal step, the vertical upper bound is updated (when verifying the update condition) and *vice versa*. When applied, this method allows the decrease of the calculation zone after each operation and, then, the computational complexity is reduced. A detailed summary of the proposed algorithm is shown below:

- ***Step 1 – Initialization:*** Initialize k_α, k_s and k_p to 1 and initialize $k_{\alpha,\max}$, $k_{s,\max}$ to n_m.
- ***Step 2 – Vertical processing:*** 1) Calculate the bubble in the column indexed by k_α. 2) Update $\alpha_i(j)$ with the computed bubble $B_{j',j}$ if $B_{j',j} < \alpha_i(j)$. 3) If the parity transition metric term $\gamma_{p,i-1}(j',j)$ in $B_{j',j}$ is $\gamma_p^{k_p}$, increment k_p.
- ***Step 3 – Update $k_{s,\max}$ and increment k_α:*** Let m be the row number of the systematic transition metric term $\gamma_{s,i-1}(j',j)$ used to compute $B_{j',j}$ in step 2. Compute a dummy bubble B' with the lowest values of $\alpha_{i-1}(j')$ and $\gamma_{p,i-1}(j',j)$ not yet used (that is with indexes $k_\alpha + 1$ and k_p due to the applied sorting) and with γ_s^{m+1}: $B' = \alpha^{k_\alpha+1} + \gamma_s^{m+1} + \gamma_p^{k_p}$. This dummy bubble sets a lower bound on the values of

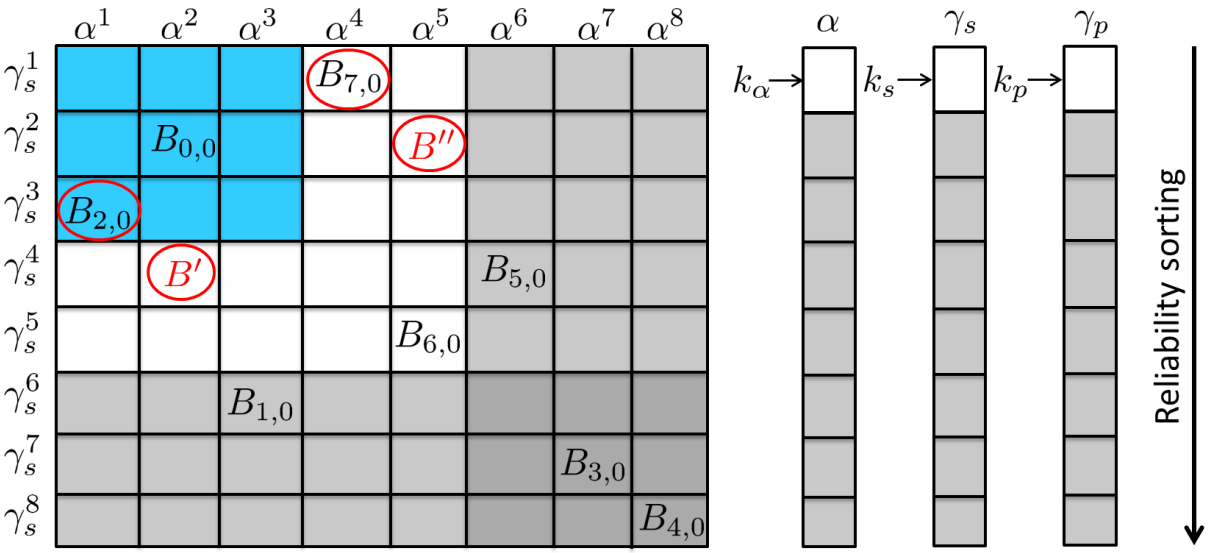

Figure 5.7: Example of sorting table used for the computation of $\alpha_i(0)$ with Eq. (5.12) for $q = 8$. All $B(j',0)$ values, $j' = 0 \cdots q-1$ are placed in different rows and columns. Min-sum processing is performed with radius $R = 3$, $k_{s,\max} = k_{\alpha,\max} = n_m = 5$. Illustration of Steps 2 and 3 (resp. Steps 4 and 5) for $k_\alpha = 1$ (resp. for $k_s = 1$), and illustration of sorted vectors α, γ_s and γ_p.

the actual bubbles located at rows with indexes greater than $m+1$. If $B' \geq \alpha_i(j)$, there is no need to continue the calculation process below this row: $k_{s,\max}$ is set to $m+1$. k_α is incremented for the next iteration.

- ***Step 4 – Horizontal processing:*** 1) Calculate the bubble in the row indexed by k_s. 2) Update $\alpha_i(j)$ with the computed bubble $B_{j',j}$ if $B_{j',j} < \alpha_i(j)$. 3) If the parity transition metric term $\gamma_{p,i-1}(j',j)$ in $B_{j',j}$ is $\gamma_p^{k_p}$, increment k_p.

- ***Step 5 – Update $k_{\alpha,\max}$ and increment k_s:*** Let n be the column number of the state metric term $\alpha_{i-1}(j')$ used to compute $B_{j',j}$ in step 4. Compute a dummy bubble B'' with the lowest values of $\gamma_{s,i-1}(j',j)$ and $\gamma_{p,i-1}(j',j)$ not yet used (that is with indexes k_s+1 and k_p due to the applied sorting) and with α^{n+1}: $B'' = \alpha^{n+1} + \gamma_s^{k_s+1} + \gamma_p^{k_p}$. This dummy bubble sets a lower bound on the values of the actual bubbles located at columns with indexes greater than $n+1$. If $B'' \geq \alpha_i(j)$, there is no need to continue the calculation process to the right of this column: $k_{\alpha,\max}$ is set to $n+1$. k_s is incremented for the next iteration.

Steps 2 to 5 are repeated until $k_s \geq k_{s,\max}$ and $k_\alpha \geq k_{\alpha,\max}$.

The proposed decoding algorithm limits the active bubbles to the ones lying inside the area bounded by $k_{\alpha,\max}$ and $k_{s,\max}$. The same process is repeated for every state $j \in [0 \cdots q-1]$ and at each trellis stage. The proposed algorithm can also be directly applied for the computation of backward state metrics and extrinsic LLRs.

5.3.3 Computational complexity analysis

The computational complexity for the calculation of the forward state metrics, backward state metrics and extrinsic LLRs is assessed in terms of the average number of ACS oper-

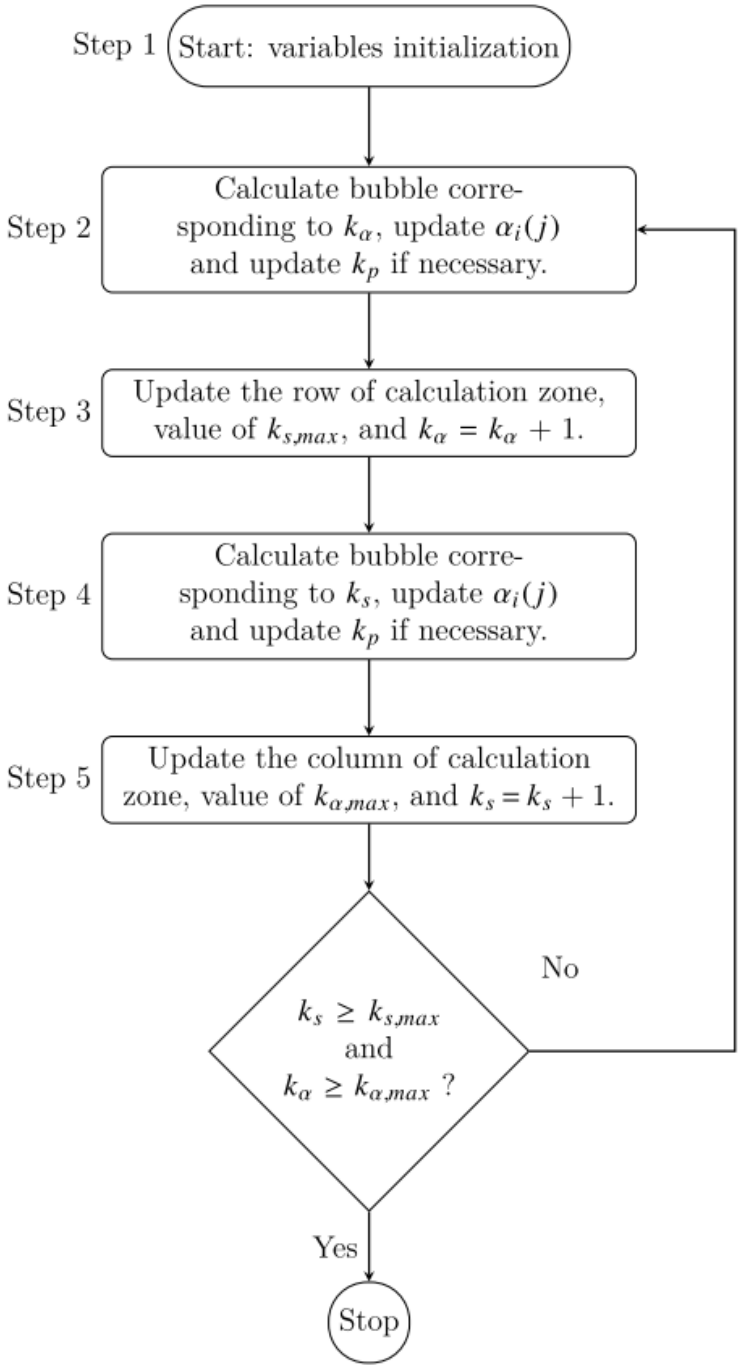

Figure 5.8: Flow chart of the proposed simplified min-sum processing algorithm.

ations executed per decoded frame. For the full Min-Log-MAP algorithm, it is calculated as follows:

$$ACS_{MLM} = 3n_{enc}.K_s.n_{it}.q^2 \tag{5.18}$$

where n_{enc} denotes the number of component encoders in the TC structure – for conventional TCs, $n_{enc} = 2$ – K_s is the number of GF(q) symbols in the messages to be encoded and n_{it} represents the number of performed decoding iterations. The factor 3 stands for the three processes needed to compute the forward and backward probabilities and the

extrinsic information.

For the proposed bubble check algorithm, the number of operations ACS_{BC} is upper bounded by:

$$ACS_{BC} = 3n_{enc}.K_s.n_{it}.2n_m.q \tag{5.19}$$

The factor 2 in Eq. (5.19) accounts for the update of the metrics in Steps 2 or 4 of the algorithm (see Section 5.3.2.2) and for the update of $k_{\max}$ in Steps 3 or 5: for each of the q encoder states, at most $2n_m$ ACS operations are needed. Therefore, from Eq. (5.18) and (5.19), it can be inferred that the proposed min-sum processing reduces the complexity of the Min-Log-MAP algorithm, provided that $n_m < \frac{q}{2}$.

5.4 Simulation and complexity results

5.4.1 Simulation settings

The error correction performance of the proposed decoder is evaluated by simulation over a Gaussian channel, in terms of Frame Error Rate (FER) versus the signal to noise ratio $\mathrm{E_s/N_0}$. NB-TCs are formed from the parallel concatenation of two NB-CCs over GF(64), where the coded symbols are mapped to a 64-QAM constellation. The considered component codes are the rate-1/2 NB-CCs $\mathrm{C_1}$ and $\mathrm{C_3}$ described in Chapter 2, Table 2.4.

A moderate message length is used, K_s = 900 GF(64) symbols, corresponding to K_b = 5400 bits. No puncturing is applied, and the overall coding rate is R = 1/3. An almost regular permutation (ARP) [60] is used as internal interleaver. The interleaver parameters and corresponding minimum spread S_{min} and girth values are given in Table 5.1.

S_{min}	Girth	P	Q	(S(0),...,S(Q-1))
30	8	137	4	(0,854,396,362)

Table 5.1: ARP interleaver parameters, GF(64), K_s = 900 symbols and R = 1/3 (ARP equation: $\Pi(i) = (Pi + S\,(i \bmod Q)) \bmod K_s$).

The turbo decoding algorithms are simulated using n_{it} = 8 decoding iterations, unless otherwise specified. 8 quantization bits are used for the representation of the input symbol LLRs and 9 bits for the forward and backward metrics.

As shown in Section 5.3.2, the storage and computational complexities depend on the values of n_m and R. In order to assess the effect of these parameters on the proposed decoding algorithm, three different configurations are proposed as shown in Table 5.2. Conf1, Conf2 and Conf3 differ in the used radius value R and in the truncation length n_m.

5.4.2 Upper bound on computational complexity

Before comparing the real computational complexity required for each configuration, Eq. (5.18) and (5.19) from Section 5.3.3 can help identify an upper bound on the number of performed ACS operations for each algorithm/configuration. Table 5.3 presents these

Configuration	Conf1	Conf2	Conf3
R	10	4	2
n_m	16	8	4

Table 5.2: Values of parameters R and n_m for the three simulated configurations.

values, as well as the percentage gain for the three simulated configurations with respect to the Min-Log-MAP algorithm.

Algorithm	MLM	Bubble Check		
		Conf1	Conf2	Conf3
# ACS$\times 10^6$/Frame	176.9	88.4	44.2	22.1
Min. complexity reduction (%)	0	50.1	75.1	87.5

Table 5.3: Upper bounds on the number of ACS operations by frame for different decoding algorithms.

5.4.3 Performance and complexity comparisons for NB-TCs using C_1 as NB-CC

Fig. 5.9 shows the error correction loss due to the reduced-complexity decoding algorithm, with respect to the original scaled Min-Log-MAP algorithm when using the component NB-CC with the worst Euclidean distance spectrum whose parameters are $(a_1, a_2, a_3) = (41, 2, 0)$. This loss varies from 0.1 to 0.9 dB at a FER equal to 10^{-3}, depending on the configuration. Two additional curves are shown, corresponding to the simulation of two binary TCs using 16-state and 64-state constituent codes and decoded with the scaled Min-Log-MAP algorithm (with scaling factors corresponding to 0.7 for the first iteration, 1 for the last one, and 0.8 for other iterations) under the same simulation conditions. The 64-state binary code is shown since it has the same number of states as the NB-TC over GF(64), while the 16-state binary code is known to have a better convergence threshold and is considered in order to assess the performance of NB-TCs in this region. We can observe that the proposed NB-TC outperforms both binary codes even when a complexity reduction is applied. The proposed code can compete with the binary TC using the same number of states, even when performing only 6 decoding iterations (curve labeled Conf3-R2-n_m4-6It) and even with the binary 16-state turbo code, which displays a lower decoding threshold than the binary 64-state code.

Fig. 5.10 compares the actual computational complexity of the different schemes in terms of the number of measured ACS operations as a function of the FER, without taking into account the complexity imposed by the sorting algorithm. A stopping criterion is incorporated into the decoding process in order to avoid performing all the iterations. The complexity reduction with respect to the classical scaled Min-Log-MAP algorithm varies from a factor 3 to almost 10, depending on the configuration. For instance, accepting a loss in the order of 0.2 dB in performance will result in a complexity reduction by a

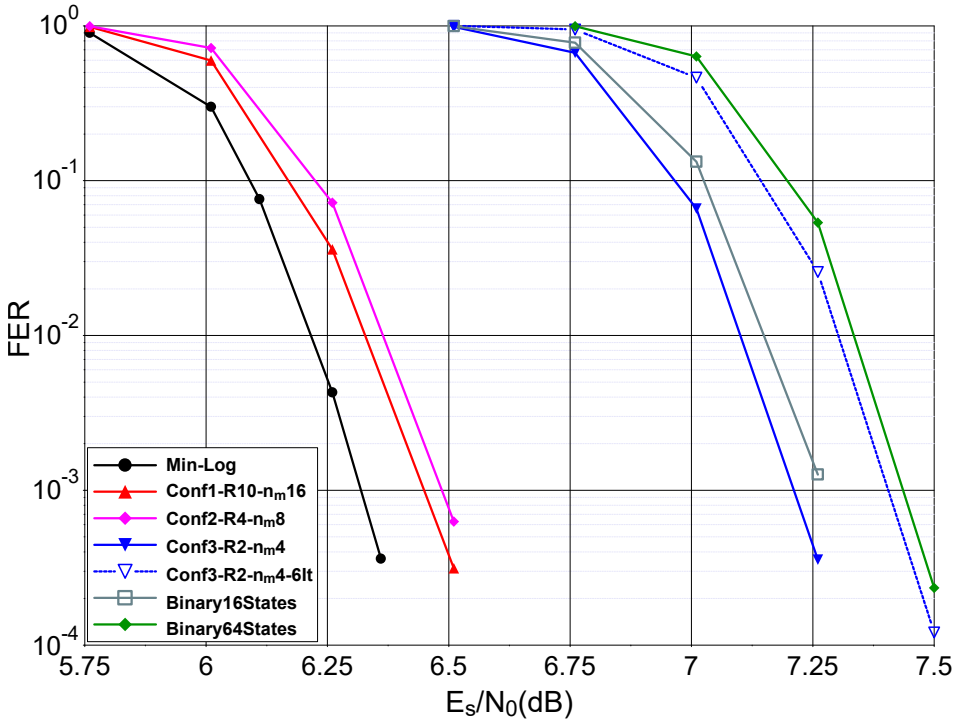

Figure 5.9: Performance comparison, in terms of FER, of the 3 low-complexity decoding configurations with Min-Log-MAP algorithm for NB-TC, 16-state and 64-state binary TC. Transmission using 64-QAM over the AWGN channel. NB-CC parameters are (a_1,a_2,a_3)=(41,2,0). Generator polynomial for binary 16-state and 64-state component codes: (1,35/23) and (1,171/133) respectively, in octal notation.

factor 6 using configuration Conf2. When comparing with the binary 16-state, the NB-TC decoded with the full scaled Min-Log-MAP algorithm is 20 times more complex and 6 times when comparing with the 64-state TCs, for a corresponding gain in performance of 0.9 dB and 1.2 dB. With the simplified configuration Conf2, a performance gain of 0.8 dB can still be achieved compared to the 16-state TC, at the price of a complexity multiplied by 4. Also a gain of 1.0 dB is achieved in comparison with the 64-state binary TC, while having the same computational complexity. Note that, although not shown in Fig. 5.10, we observed that the NB-TC is able to achieve lower error floors than the 16-state binary TC.

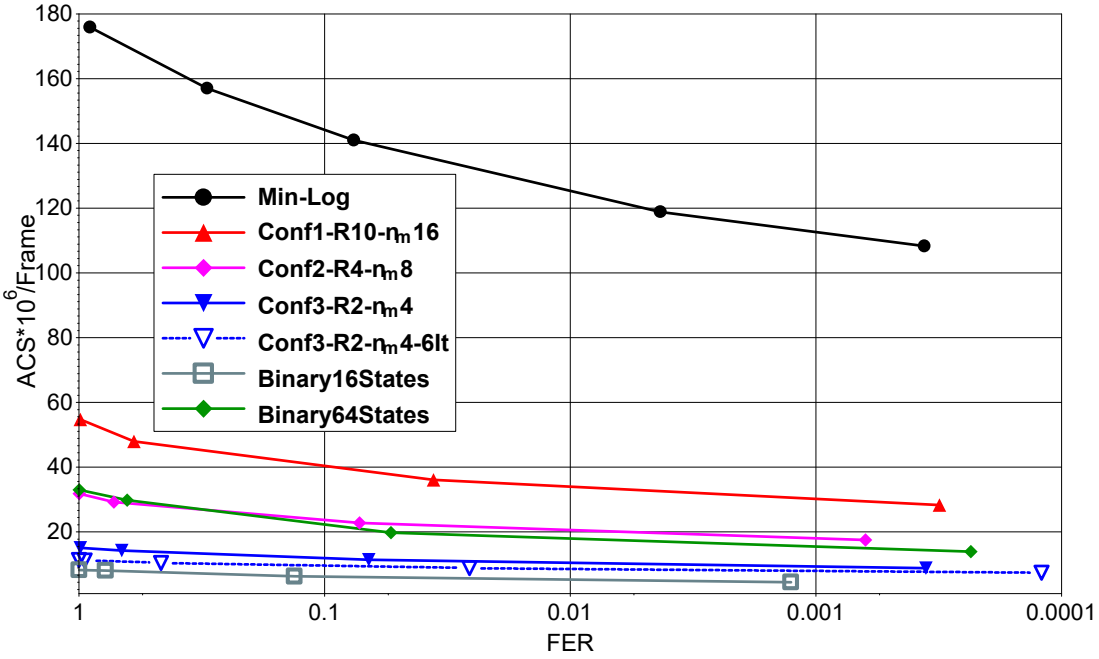

Figure 5.10: Comparison of the computational complexity in terms of ACS operations as a function of the achieved FER for the codes simulated in Fig. 5.9.

Moreover, even when requiring a higher number of operations, the NB decoder should increase the decoded data throughput, compared to the binary TC, thanks to symbol-based processing that provides $\log_2(q)$ bits when decoding a trellis section. Therefore, more in-depth comparisons should be performed at the same throughput level which would require considering high order radix-based decoders for binary codes.

5.4.4 Performance and complexity comparisons for NB-TCs using C_3 as NB-CC

The performance and complexity comparisons were also conducted when using the component NB-CC with the best Euclidean distance spectrum whose parameters are (a_1,a_2,a_3) = (31,5,18). In this context, an additional penalty at low SNR is observed, while lower error floor region can be achieved. In Fig. 5.11, the performance comparison is depicted and the conclusions are similar to the previous case with code C_1. As expected, the error correction capabilities decrease when the complexity is reduced. When performing at 10^{-3} of FER, Conf1 shows a performance loss lower than 0.1 dB when compared to the

performance offered by the classical scaled Min-Log-MAP decoding algorithm. Performance penalties slightly increase when applying the complexity reduction corresponding to Conf2, with a performance gap of 0.15 dB compared to scaled Min-Log-MAP decoding. When aiming at a higher complexity reduction, with configuration Conf3, a loss in the order of 1.2 dB can be observed with respect to the reference curve.

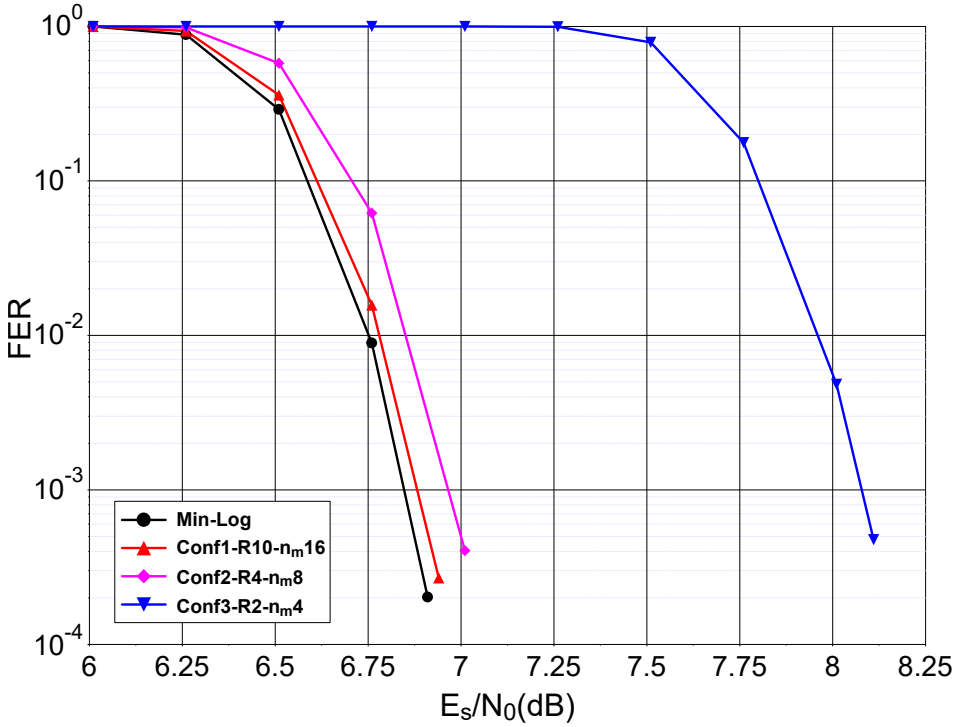

Figure 5.11: Performance comparison, in terms of FER, of the 3 low-complexity decoding configurations with Min-Log-MAP algorithm for NB-TC. Transmission using 64-QAM over the AWGN channel. NB-CC parameters are (a_1,a_2,a_3)=(31,5,18).

Fig. 5.12 depicts the actual number of ACS operations as function of the FER for the scaled Min-Log-MAP decoding algorithm in comparison with the three proposed configurations for the reduced complexity decoding algorithm. Similarly to the case when C_1 is used, the proposed decoding algorithm shows a reduced number of ACS operations with respect to the classical scaled Min-Log-MAP algorithm. This reduction depends on the adopted configuration, and entails performance losses depending on the targeted complexity.

5.4.5 Discussion

We showed that the values of n_m and R have a high impact on the storage requirements and on the computational complexity of the decoder. When reducing the computation and memory complexity, some silicon area is saved and less power is consumed but there is a price to pay in terms of error correction performance loss.

In conclusion, the choice of the best configuration will depend on different criteria. When accepting high performance penalties, very high complexity reductions can be achieved while still offering better performance than binary codes. Indeed, Conf3 with the worst NB-CC C_1 can perform very closely to binary 16 and 64-state TCs, while having

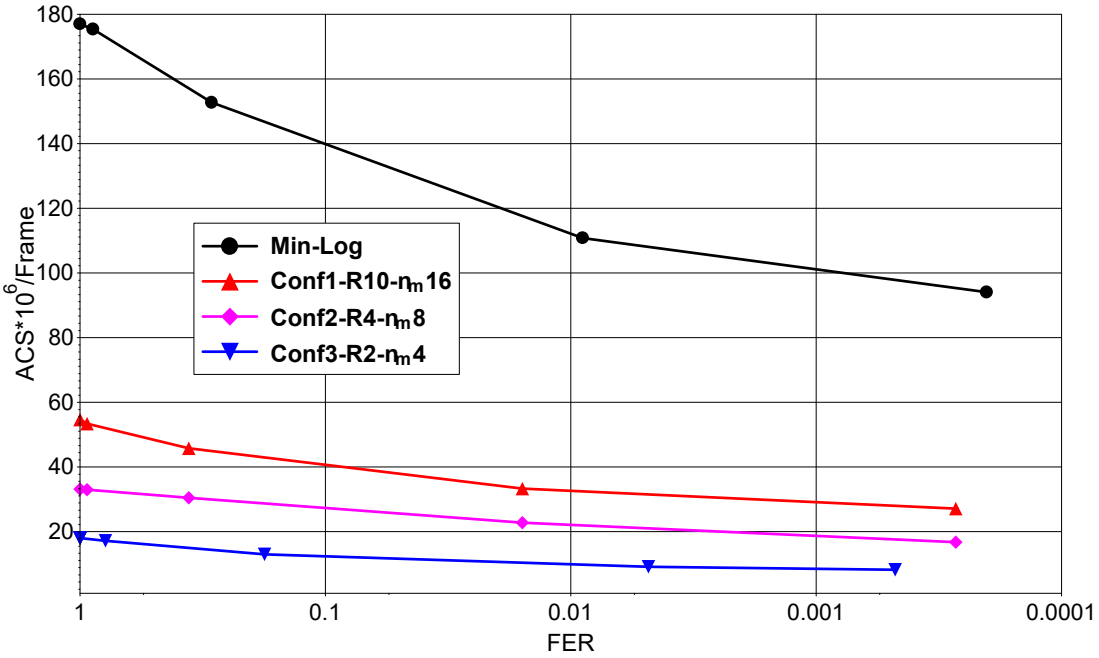

Figure 5.12: Computational complexity comparison in terms of ACS operations as a function of the achieved FER for the NB-TCs simulated in Fig. 5.11.

comparable or even lower computational complexity. Memory savings mainly depend on the value of n_m. With values of n_m as low as 4, as in Conf3, the hardware implementation of GF(64) TC decoders becomes practically feasible with nowadays semiconductor technology. When the error correction performance is the main constraint, the performance gains come at the price of a non-negligible increase of the decoder complexity, as seen with configurations such as Conf1 and Conf2. This compromise between complexity and performance should be carefully weighed for each application independently in order to define the best code with the best decoding algorithm parameters.

5.5 Conclusion

In this chapter, a new decoding algorithm is proposed in the context of NB-TCs over finite fields GF(q). The designed algorithm is a largely modified variant of the bubble check sorting, which is mainly used in the low-complexity decoding algorithms of NB-LDPC codes. Storage complexity is reduced by vector truncation. Computational complexity is reduced by the bubble check-like decoder. The same algorithm is applied for the state metric recursive calculations as well as for the extrinsic computations.

The proposed algorithm is shown to achieve high complexity reductions at the price of error correction performance penalties. The trade-off between complexity and performance depends on the constraints of the application. It was demonstrated that, when $n_m < \frac{q}{2}$, the proposed algorithm has a lower computational complexity than the Min-Log-MAP algorithm. If a performance loss as high as 1 dB is tolerated, the computational complexity can be divided by a factor in the order of 10, compared to the Min-Log-MAP. Other more favorable performance/complexity tradeoffs can also be achieved.

The low-complexity decoding algorithm proposed in this chapter was published and presented at the 10$^{\text{th}}$ International Symposium on Turbo Codes and Iterative Information

Processing (ISTC 2018):

R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah, "Low-complexity decoders for non-binary turbo codes", in 10^{th} International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018), Hong Kong, Dec. 2018.

Conclusions and future works

In this thesis, we have studied a new family of error correcting codes based on Non-Binary Turbo Codes (NB-TCs). All the component building blocks are addressed in this work.

First, we began by defining a new structure of NB-CCs. The proposed constituent encoder is based on a new single memory element structure, which is found to enhance the lowest Euclidean distance of the well-known accumulator structure by more than 25%. These NB-CCs are associated with different constellation orders, and are found to guarantee better performance than existing binary and non-binary convolutional codes.

Then, we focused on defining the permutation function that can offer the best coding gains for the resulting NB-TC. It was identified that Almost Regular Permutation (ARP) interleavers are the best suited in this case since they can simultaneously satisfy defined criteria related to the spread and girth values at design time on the fly. Moreover, This type of interleaver can also support several newly proposed criteria introduced to reduce correlation in the iterative decoding.

For rate adaptation, we have studied different types of puncturing that can be imagined when using symbol encoding. Constrained by the constellation mapping and order, we were able to propose bit puncturing schemes for NB-TCs that offer improved threshold performance by more than 0.5 dB for high coding rates.

With a largely improved asymptotic performance, it was important to find a way to predict NB-TCs behavior at low error rates without having to resort to long simulation runs. While far from being a straightforward extension of the binary case, the derivation of the union bounds for NB-TCs provides the means to do just that. However, it requires the estimate of, at least, a truncated distance spectrum of the Euclidean distances of the NB-TC. In Chapter 3, we propose the method to estimate this truncated spectrum and therefore to compute these bounds. They have proven to be accurate for several application cases including the combination of different codes with binary and high order constellations.

To improve further low error rate performance, we proposed a new structure of NB-TCs by adding a transformation block before the second constituent encoder. When well-defined, this block helps breaking error patterns that result in the lowest distance of the considered codes. Error floors are found to be lowered with more than two decades, without penalizing the performance at high error rates.

Finally, a low-complexity decoding algorithm was derived. The proposal consists in reducing the high storage requirements, as well as the high number of operations. It results in drastically reducing the complexity of NB-TCs by a factor between 4 and 6 of the number of add-compare-select units, with a small impact on the error correction performance of the code.

Performance evaluation and comparisons of the obtained codes with existing codes in the literature are conducted under different scenarios. The designed NB-TCs are shown

to offer better performance, when compared with the best state-of-the-art binary and non-binary codes, for different frame sizes, coding rates, and constellation orders.

Future works

The work presented in this thesis showed that NB-TCs can present several advantages for future communication standards targeting reliable communications, with very high throughput. Yet, not all the topics related to this structure have been thoroughly explored.

We began this work by defining a low-complexity memory-1 structure of NB-CCs. This structure offers a good trade-off between the error correction capability and the decoding complexity. After finding a reduced-complexity decoding algorithm, one should further investigate the design of these NB-CCs. This work can be extended to design memory-2 NB-CCs, that may achieve better error correction capabilities at the detriment of the resulting complexity. Such a study requires the adaptation of the distance spectrum calculation algorithm, since in this case, longer Diverging and Converging (DC) sequences start to impact performance. Avoiding short DC sequences may also be obtained by time-variant trellis codes, where the non-binary coefficients, defining the trellis of the code, vary with time. The optimization of these coefficients should be studied jointly with the resulting trellises.

For rate flexibility, puncturing techniques can be explored further by targeting communication systems that allow rate compatibility. To this end, the puncturing functions should be chosen accordingly. Now that union bounds can be computed, the interleaver design process can be updated to compute union bounds as a final selection phase between candidate interleavers that satisfy girth, spread and the proposed new constraints for low correlation in decoding.

We have proposed in Chapter 4 to add a bijective transformation block to the structure of NB-TCs. This block was designed to avoid retrieving the same competing sequences at both decoders. A further optimization of this block can be considered, by trying to find an optimal solution for the proposed connections. This solution should be based on the definition of a mathematical problem that aims at enhancing the minimum distance of the code.

On the other hand, a hardware implementation should be considered for the proposed decoder in Chapter 5, which is shown to drastically reduce the computational complexity of the non-binary decoder without hardly penalizing the excellent performance of these codes. The extension of this algorithm to even higher radix orders can be envisioned for achieving even higher throughput levels. And finally on another note, the proposed decoder can be extended to the binary TCs, targeting an increase in the throughput.

List of publications

Journal papers

- R. Klaimi, C. Abdel Nour, C. Douillard and J.Farah "Union bound evaluation for Non-Binary Turbo Codes", submitted to IEEE Trans. on Commun.
- R. Klaimi, C. Abdel Nour, C. Douillard and J.Farah "Enhancing performance of non-binary turbo codes via symbol transformation", to be submitted to IEEE Trans. on Commun.

Conference papers

- R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah, "Design of Low-Complexity Convolutional Codes over GF(q)", IEEE Global Commun. Conf.: Communication Theory (Globecom2018 CT), Abu Dhabi, United Arab Emirates, Dec. 2018.
- R. Garzón Bohórquez, R. Klaimi, C. Abdel Nour and C. Douillard, "Mitigating Correlation Problems in Turbo Decoders", in 10^{th} International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018), Hong Kong, Dec. 2018.
- R. Klaimi, C. Abdel Nour, C. Douillard and J. Farah, "Low-complexity decoders for non-binary turbo codes", in 10^{th} International Symposium on Turbo Codes and Iterative Information Processing (ISTC 2018), Hong Kong, Dec. 2018.

Patent filing

- R. Klaimi, C. Abdel Nour and C. Douillard "*Procédé de génération d'un signal mettant en oeuvre un turbo-codeur, dispositif et programme d'ordinateur correspondants* (in French)", Patent application FR 1873615, Dec. 2018.

Bibliography

[1] C. Mobile, "Cisco visual networking index: Global mobile data traffic forecast update 2017–2022 white paper," 2019.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, Oct 1948.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE Int. Conf. Commun., (ICC'93)*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[4] D. J. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[5] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inform. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[6] G. Liva, L. Gaudio, T. Ninacs, and T. Jerkovits, "Code design for short blocks: A survey," *arXiv preprint arXiv:1610.00873*, 2016.

[7] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, 1996.

[8] C. Berrou, K. Amis Cavalec, M. Arzel, A. Graell I Amat, M. Jezequel, C. Langlais, R. Le Bidan, Y. Saouter, E. Maury, G. Battail, E. Boutillon, A. Glavieux, Y. O. C. Mouhamedou, S. Saoudi, C. Laot, S. Kerouedan, F. Guilloud, and C. Douillard, *Codes and Turbo Codes*, ser. IRIS international series. Paris: Springer-Verlag, 2010.

[9] S. Crozier, "New high-spread high-distance interleavers for turbo-codes," in *20th Biennial Symp. on Commun.*, Queen's University, Kingston, Ontario, Canada, 2000, pp. 3–7.

[10] Y. Saouter, "Selection procedure of turbocode parameters by combinatorial optimization," in *6th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, Sept 2010, pp. 156–160.

[11] O. F. Acikel and W. E. Ryan, "Punctured turbo-codes for BPSK/QPSK channels," *IEEE Tran. on Commun.*, vol. 47, no. 9, pp. 1315–1323, 1999.

[12] R. Garzón Bohórquez, C. Abdel Nour, and C. Douillard, "On the equivalence of interleavers for turbo codes," *IEEE Wireless Commun. Lett.*, vol. 4, no. 1, pp. 58–61, Feb. 2015.

[13] R. Garzón-Bohórquez, C. A. Nour, and C. Douillard, "Protograph-based interleavers for punctured turbo codes," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 1833–1844, 2018.

[14] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. Telecommun.*, vol. 8, no. 2, pp. 119–125, 1997.

[15] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[16] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, 1998.

[17] G. Böcherer, P. Schulte, and F. Steiner, "Probabilistic shaping and forward error correction for fiber-optic communication systems," *J. of Lightw. Technol.*, vol. 37, no. 2, pp. 230–244, Jan 2019.

[18] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[19] M. C. Davey and D. J. MacKay, "Low density parity check codes over GF (q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, June 1998.

[20] D. J. MacKay, "Optimizing sparse graph codes over GF (q)," *online: http://www.inference.phy.cam.ac.uk/mackay/CodesGallager.html*, 2003.

[21] C. Poulliat, M. Fossorier, and D. Declercq, "Using binary images of non binary LDPC codes to improve overall performance," in *IEEE Int. Symp. on Turbo Codes and Related Topics*, Munich, Germany, 2006, pp. 1–6.

[22] ——, "Design of non binary LDPC codes using their binary image: algebraic properties," in *IEEE Int. Symp. on Inform. Theory*, Seattle, WA, USA, July 2006, pp. 93–97.

[23] M. Golay, "Notes on digital coding," *Proc. I.R.E.*, vol. 37, p. 637, 1949.

[24] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[25] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, pp. 37–46, 1955.

[26] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68–79, 1960.

[27] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[28] ETSI, "Digital Video Broadcasting (DVB): interaction channel for satellite distribution systems," EN 301 790 (V1.3.1), March 2003.

[29] ——, "LTE Envolved Universal Terrestrial Radio Access(E-UTRA): Multiplexing and channel coding," TS 136 212 (V10.0.0), January 2011.

[30] J. Hokfelt, O. Edfors, and T. Maseng, "Turbo codes: Correlated extrinsic information and its impact on iterative decoding performance," in *IEEE 49th Vehicular Technology Conf.*, vol. 3, Houston, TX, USA, 1999, pp. 1871–1875.

[31] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan, "The advantages of non-binary turbo codes," in *IEEE Inform. Theory Workshop*, Cairns, Queensland, Australia, 2001, pp. 61–63.

[32] H. Song and J. Cruz, "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," *IEEE Trans. Magnetics*, vol. 39, no. 2, pp. 1081–1087, 2003.

[33] E. Zehavi, "8-psk trellis codes for a rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.

[34] A. Alvarado, "On bit-interleaved coded modulation with QAM constellations," Master's thesis, Chalmers University of Technology, 2008.

[35] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov 1996.

[36] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[37] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2, dc)-LDPC codes over GF(q) using their binary images," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, Oct. 2008.

[38] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *IEEE Global Commun. Conf.*, vol. 3, St. Louis, MO, USA, 2005, pp. 5–pp.

[39] D. Declercq, M. Colas, and G. Gelle, "Regular GF(2^q)-LDPC modulations for higher order QAM-AWGN channels," in *Int. Symp. on Infor. Theory and its Applications ISITA*, Parma, Italy, 2004.

[40] A. Abdmouleh, E. Boutillon, L. Conde-Canencia, C. A. Nour, and C. Douillard, "A new approach to optimise non-binary LDPC codes for Coded Modulations," in *Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, 2016, pp. 295–299.

[41] ——, "On signal space diversity for non binary coded modulation schemes," in *IEEE Int. Conf. on Telecommunications (ICT)*, Thessaloniki, Greece, 2016, pp. 1–5.

[42] A. J. Viterbi and J. K. Omura, *Principles of digital communication and coding*. McGraw-Hill, New York, 1979.

[43] F. Kienle, N. Wehn, and H. Meyr, "On complexity, energy-and implementation-efficiency of channel decoders," *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3301–3310, 2011.

[44] G. Liva, S. Scalise, E. Paolini, and M. Chiani, "Turbo codes based on time-variant memory-1 convolutional codes over Fq," in *IEEE Int. Conf. Commun.*, Kyoto, Japan, June 2011, pp. 1–6.

[45] G. Liva, E. Paolini, B. Matuz, S. Scalise, and M. Chiani, "Short turbo codes over high order fields," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2201–2211, June 2013.

[46] Y. Zhao, "Convolutional codes defined in GF(q) combine with PNC over impulsive noise channels," in *IEEE Int. Conf. for Students on Applied Engineering (ICSAE)*, Newcastle upon Tyne, UK, Oct. 2016, pp. 142–146.

[47] Y. Zhao, M. Johnston, C. Tsimenidis, and L. Chen, "Non-binary turbo-coded physical-layer network coding on impulsive noise channels," *Electron. Lett.*, vol. 52, no. 24, pp. 1984–1986, 2016.

[48] W. Abd-Alaziz, M. Johnston, and S. Le Goff, "Non-binary turbo codes on additive impulsive noise channels," in *the 10th Int. Symp. Commun. Systems, Networks and Digital Signal Process. (CSNDSP)*, Prague, Czech Republic, July 2016, pp. 1–5.

[49] W. Abd-Alaziz, Z. Mei, M. Johnston, and S. Le Goff, "Non-binary turbo-coded OFDM-PLC system in the presence of impulsive noise," in *the 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos, Greece, 2017, pp. 2576–2580.

[50] N. Tsgrangra, "Evolved universal terrestrial radio access (e-utra); multiplexing and channel coding," *3rd Generation Partnership Project (3GPP), vol. TS*, vol. 36, 2009.

[51] D. Divsalar and F. Pollara, "Multiple turbo codes for deep-space communications," *The Telecommunications and Data Acquisition Progress Report 42-121, May 1995.*

[52] J. Hokfelt, O. Edfors, and T. Maseng, "A turbo code interleaver design criterion based on the performance of iterative decoding," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 52–54, 2001.

[53] H. R. Sadjadpour, M. Salehi, N. Sloane, and G. Nebe, "Interleaver design for short block length turbo codes," in *IEEE Int. Conf. on Commun. 2000*, vol. 2, New Orleans, LA, USA, June 2000, pp. 628–632.

[54] Y. Saouter, "Selection procedure for binary turbocode permutation parameters," in *5th Int. Symp. on Turbo Codes and Related Topics*, Lausanne, Switzerland, Sept 2008, pp. 272–276.

[55] D. Divsalar and F. Pollara, "Turbo codes for deep-space communications," *The Telecommunications and Data Acquisition Progress Report 42-120, October–December 1994*, pp. 29–39.

[56] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[57] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *IEEE Glob. Commun. Conf. (GLOBECOM)*, vol. 3. San Francisco, CA, USA: IEEE, 1994, pp. 1298–1303.

[58] E. Boutillon and D. Gnaedig, "Maximum spread of d-dimensional multiple turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1237–1242, 2005.

[59] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," *The Telecommunications and Data Acquisition Progress Report 42-122*, pp. 56–65, Aug 1995.

[60] C. Berrou, Y. Saouter, C. Douillard, S. Kerouédan, and M. Jézéquel, "Designing good permutations for turbo codes: towards a single model," in *IEEE Int. Conf. Commun.*, vol. 1, Paris, France, 2004, pp. 341–345.

[61] S. Crozier and P. Guinand, "High-performance low-memory interleaver banks for turbo-codes," in *IEEE 54th Vehicular Technology Conf.*, vol. 4, Atlantic City, NJ, USA, Oct 2001, pp. 2394–2398.

[62] J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 101–119, 2005.

[63] IEEE, "IEEE standard for local and metropolitan area networks, Part 16: Air interface for fixed and mobile broadband wireless access systems," IEEE Std 802.16-2004/Cor 1-2005, Feb. 2006.

[64] ETSI, "Digital Video Broadcasting (DVB): second generation DVB interactive satellite system (DVB-RCS2): Part 2: Lower layers for satellite standard," EN 301 545-2 (V1.1.1), January 2012.

[65] L. Perez, J. Seghers, and D. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, 1996.

[66] B. Vucetic and J. Yuan, *Turbo codes: principles and applications*. Springer Science & Business Media, 2012, vol. 559.

[67] C. Berrou, S. Vaton, M. Jezequel, and C. Douillard, "Computing the minimum distance of linear codes by the error impulse method," in *IEEE Global Telecommun. Conf.*, vol. 2, Taipei, Taiwan, 2002, pp. 1017–1020.

[68] R. Garello and A. Vila-Casado, "The all-zero iterative decoding algorithm for turbo code minimum distance computation," in *IEEE Int. Conf. Commun.*, vol. 1, Paris, France, June 2004, pp. 361–364.

[69] S. Crozier, P. Guinand, and A. Hunt, "Estimating the minimum distance of turbo-codes using double and triple impulse methods," *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 631–633, 2005.

[70] ——, "Estimating the minimum distance of large-block turbo codes using iterative multiple-impulse methods," in *IEEE Int. Symp. on Turbo Codes and Related Topics*, Munich, Germany, 2006, pp. 1–6.

[71] S. Crozier, K. Gracie, and R. Kerr, "Estimating the minimum distance of large-block turbo codes using the event impulse method," in *Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, 2010, pp. 439–443.

[72] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[73] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE Int. Conf. Commun., (ICC'95)*, vol. 2, Seattle, USA, 1995, pp. 1009–1013.

[74] C. Douillard and C. Berrou, "Turbo codes with rate-m/(m+ 1) constituent convolutional codes," *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1630–1638, 2005.

[75] C. Berrou and M. Jezequel, "Non-binary convolutional codes for turbo coding," *Electron. Lett.*, vol. 35, no. 1, pp. 39–40, 1999.

[76] C. Berrou, "Some clinical aspects of turbo codes," in *Int. Symp. on Turbo Codes and related topics*, Brest, France, 1997, pp. 26–31.

[77] J. L. Massey and T. Mittelholzer, "Convolutional codes over rings," in *4th Joint Swedish Soviet Int. Workshop Inform. Theory*, Gotland, Sweden, 1989, pp. 14–18.

[78] R. Baldini Filho and P. G. Farrell, "Coded modulation with convolutional codes over rings," in *EUROCODE'90*. Springer, 1991, pp. 271–280.

[79] R. Johannesson, Z.-X. Wan, and E. Wittenmark, "Some structural properties of convolutional codes over rings," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 839–845, 1998.

[80] F. Fagnani and S. Zampieri, "System-theoretic properties of convolutional codes over rings," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2256–2274, 2001.

[81] T. Konishi, "A coded modulation scheme for 64-QAM with a matched mapping," in *IEEE Int. Symp. Inform. Theory and its Applications (ISITA)*, Oct. 2014, pp. 191–195.

[82] R. A. Carrasco and M. Johnston, *Non-binary error control coding for wireless communication and data storage*. Wiley, 2008.

[83] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[84] S. Benedetto and E. Biglieri, *Principles of digital transmission: with wireless applications*. Springer Science & Business Media, 1999.

[85] M. K. Simon, B. Levitt, J. Omura, and R. Scholtz, "Spread spectrum communications. volume 1, 2 & 3," *NASA STI/Recon Technical Report A*, vol. 87, 1985.

[86] D. Divsalar and M. Simon, "Trellis coded modulation for 4800-9600 bits/s transmission over a fading mobile satellite channel," *IEEE J. Select. Areas in Commun.*, vol. 5, no. 2, pp. 162–175, Feb. 1987.

[87] P. Frenger, P. Orten, and T. Ottosson, "Convolutional codes with optimum distance spectrum," *IEEE Commun. Lett.*, vol. 3, no. 11, pp. 317–319, 1999.

[88] N. Biggs, "Minimal regular graphs with given girth," in *Algebraic graph theory*. New York, NY, USA: Cambridge University Press, 1974, pp. 180–190.

[89] R. Garzón Bohórquez, C. Abdel Nour, and C. Douillard, "Improving turbo codes for 5G with parity puncture-constrained interleavers," in *Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, Sept. 2016, pp. 151–155.

[90] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate-repeat-accumulate codes," *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 692–702, 2007.

[91] S. Tong, H.-j. Zheng, and B.-m. Bai, "Precoded turbo code within 0.1 db of shannon limit," *Electron. Lett.*, vol. 47, no. 8, pp. 521–522, 2011.

[92] R. G. Bohorquez, C. A. Nour, and C. Douillard, "Precoding techniques for turbo codes," in *21th European Wireless Conf.* Budapest, Hungary: VDE, May 2015, pp. 1–6.

[93] 3GPP, "Multiplexing and channel coding. 3rd Generation Partnership Project," TS 38.212, v15.0.0, Release 15, 2018.

[94] D. Hui, S. Sandberg, Y. Blankenship, M. Andersson, and L. Grosjean, "Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE," *IEEE Veh. Tech. Magazine*, vol. 13, no. 4, pp. 60–69, 2018.

[95] R. Maunder, "Matlab simulations of the encoder and decoder for the New Radio LDPC code from 3GPP Release 15," Nov 2018. [Online]. Available: https://github.com/robmaunder/ldpc-3gpp-matlab

[96] Lab-STICC, "NB-LDPC matrices and simulation results." [Online]. Available: http://www-labsticc.univ-ubs.fr/nb-ldpc/

[97] D. Feng, H. Xu, Q. Zhang, Q. Li, Y. Qu, and B. Bai, "Nonbinary LDPC-Coded Modulation System in High-Speed Mobile Communications," *IEEE Access*, vol. 6, pp. 50 994–51 001, 2018.

[98] I. Gutierrez, G. Bacci, J. Bas, A. Bourdoux, H. Gierszal, A. Mourad, and S. Pleftschinger, "Davinci non-binary ldpc codes: Performance and complexity assessment," in *IEEE Future Network & Mobile Summit*, Florence, Italy, 2010, pp. 1–8.

[99] R. Maunder, "Matlab simulations of the encoder and SCL decoder for the New Radio polar code from 3GPP Release 15," Aug 2018. [Online]. Available: https://github.com/robmaunder/polar-3gpp-matlab

[100] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF (q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, 2007.

[101] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1365–1375, 2010.

[102] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[103] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity, low-memory EMS algorithm for non-binary LDPC codes," in *IEEE Int. Conf. on Commun.*, Glasgow, UK, 2007, pp. 671–676.

[104] L. Conde-Canencia, A. Al-Ghouwayel, and E. Boutillon, "Complexity comparison of non-binary LDPC decoders," in *ICT Mobile Summit*, 2009, pp. 1–8.

[105] E. Boutillon and L. Conde-Canencia, "Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders," *Electron. Lett.*, vol. 46, no. 9, pp. 633–634, 2010.

[106] ——, "Simplified check node processing in nonbinary LDPC decoders," in *6th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, Sept 2010, pp. 201–205.

[107] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[108] N. Wiberg, "Codes and decoding on general graphs," PhD dissertation, Linkoping University, Linkoping, Sweden, 1996.

[109] M. P. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, 1999.

[110] J. Chen and M. P. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, 2002.

[111] J. Zhao, F. Zarkeshvari, and A. Banihashemi, "On implementation of min-sum algorithm and its modifications for decoding LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549–554, 2005.

[112] J. Vogt and A. Finger, "Improving the Max-Log-MAP turbo decoder," *Electron. Lett.*, vol. 36, no. 23, pp. 1937–1939, Nov 2000.

[113] V. Savin, "Min-Max decoding for non binary LDPC codes," in *IEEE Int. Symp. on Inform. Th.*, Toronto, ON, Canada, 2008, pp. 960–964.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Etude de turbocodes non binaires pour les futurs systèmes de communication et de diffusion

Mots clés : codes convolutifs, turbocodes, corps de Galois, code non binaire, modulation codée, entrelaceur ARP, réduction de complexité.

Résumé : Les systèmes de téléphonie mobile de 4ème et 5ème générations ont adopté comme techniques de codage de canal les turbocodes, les codes LDPC et les codes polaires binaires. Cependant, ces codes ne permettent pas de répondre aux exigences, en termes d'efficacité spectrale et de fiabilité, pour les réseaux de communications futurs (2030 et au-delà), qui devront supporter de nouvelles applications telles que les communications holographiques, les véhicules autonomes, l'internet tactile … Un premier pas a été fait il y a quelques années vers la définition de codes correcteurs d'erreurs plus puissants avec l'étude de codes LDPC non binaires, qui ont montré une meilleure performance que leurs équivalents binaires pour de petites tailles de code et/ou lorsqu'ils sont utilisés sur des canaux non binaires. En contrepartie, les codes LDPC non binaires présentent une complexité de décodage plus importante que leur équivalent binaire. Des études similaires ont commencé à émerger du côté des turbocodes. Tout comme pour leurs homologues LDPC, les turbocodes non binaires présentent d'excellentes performances pour de petites tailles de blocs. Du point de vue du décodage, les turbocodes non binaires sont confrontés au même problème d'augmentation de la complexité de traitement que les codes LDPC non binaire. Dans cette thèse nous avons proposé une nouvelle structure de turbocodes non binaires en optimisant les différents blocs qui la constituent. Nous avons réduit la complexité de ces codes grâce à la définition d'un algorithme de décodage simplifié. Les codes obtenus ont montré des performances intéressantes en comparaison avec les codes correcteur d'erreur de la littérature.

Title : Study of non-binary turbo codes for future communication and broadcasting systems

Keywords : convolutional codes, turbo codes, Galois fields, non-binary codes, coded modulation, ARP interleaver, complexity reduction.

Abstract: Nowadays communication standards have adopted different binary forward error correction codes. Turbo codes were adopted for the long term evolution standard, while binary LDPC codes were standardized for the fifth generation of mobile communication (5G) alongside with the polar codes. Meanwhile, the focus of the communication community is shifted towards the requirement of beyond 5G standards. Networks for the year 2030 and beyond are expected to support novel forward-looking scenarios, such as holographic communications, autonomous vehicles, massive machine-type communications, tactile Internet… To respond to the expected requirements of new communication systems, non-binary LDPC codes were defined, and they are shown to achieve better error correcting performance than the binary LDPC codes. This performance gain was followed by a high decoding complexity, depending on the field order. Similar studies emerged in the context of turbo codes, where the non-binary turbo codes were defined, and have shown promising error correcting performance, while imposing high complexity. The aim of this thesis is to propose a new low-complex structure of non-binary turbo codes. The constituent blocks of this structure were optimized in this work, and a new low-complexity decoding algorithm was proposed targeting a future hardware implementation. The obtained results are promising, where the proposed codes are shown to outperform existing binary and non-binary codes from the literature.