DIWINE

Contract No. FP7-ICT-318177

Terminal node processing for advanced scenarios D4.03

Contractual date:	M36
Actual date:	M36
Authors:	Tomáš UŘIČÁŘ, Pavel PROCHÁZKA, Jan SÝKORA, Jan HEJTMÁNEK, Kostas RAMANTAS, John S. VARDAKAS, Umberto SPAGNOLINI, Ali PARICHEHREH, Alister G. BURR, Mehdi MORTAZAWI MOLU, Yi WANG
Participants:	CTU, IQU, PdM, UoY
Work package:	WP4
Security:	Public
Nature:	Report
Version:	1.0
Number of pages:	240

Abstract

This deliverable presents a comprehensive summary of the selected algorithms implemented in hardware and system level demonstrations. In addition to this, it introduces a set of novel techniques and algorithms developed in the last stage of the project.

Keywords

Wireless physical layer network coding, hierarchical decode and forward.

Executive summary

This report is the final version of the deliverable on “Terminal node processing for advanced scenarios”. It serves two main purposes: (i) it provides a comprehensive technical overview of selected physical layer (PHY) techniques and algorithms, which have been successfully implemented by the hardware (HW) and system level simulator (SLS) demonstrators (as reported in the WP5 deliverables), and (ii) it summarises additional promising PHY techniques and algorithms developed in the last stage of the project.

Partners have collaborated to support the demonstration activities (WP5), building upon the scenarios agreed in WP2 to implement selected algorithms developed in WP4. In addition, this report introduces several novel algorithms and techniques which were devised by project partners in the last year of the project. Due to time constraints, these solutions could not be integrated on the HW/SLS platforms.

Contents

Abbreviations

2-WRC	2-Way Relay Channel
AF	Amplify and Forward
APP	A-Posteriori Probability
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek and Raviv (algorithm)
BER	Bit Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
CA	Cloud Access
CaF/CF	Compute and Forward
CC	Constellation-constrained
CCD	Construction by Code Formula
cdf	cumulative density function
CIMC	Critical Industrial Monitoring and Control
CLDLC	Complex Low Density Lattice Codes
CN	Check Node(s)
CSI	Channel State Information
EDC	Elementary Divisor Construction
EXIT	Extrinsic Information Transfer Chart
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
f.g.	finitely generated
FTN	Faster Than Nyquist
GF	Galois Field
GMR	Gaussian Mixture Reduction
H-BC	Hierarchical Broadcast Channel
H-IFC	Hierarchical Interference Cancellation
H-MAC	Hierarchcial Multiple Access Channel
H-NTF	Hierarchical Network Transfer Function
H-PEP	Hierarchical Pair-wise Error Probability
H-SCFD	Hierarchical Successive CaF Decoding
HDF	Hierarchical Decode and Forward
HI	Hierarchical Information
HIC	Hierarchical Interference Cancellation
HNC	Hierarchical Network Code
HSC	Hypercube Shaping Capacity

HSI	Hierarchical Side Information
HW	Hardware
i.i.d.	independent, identically distributed
IFC	Interference Cancellation
IC	Interference Canceller
ISI	Inter Symbol Interference
IMSD	Iterative Multi-stage Decoder
LD	Lattice Decoding
LDLC	Low Density Lattice Codes
LDPC	Low Density Parity Check (code)
LER	Largest Eigenmode Relaying
LIF	Layered Integer Forcing
LLR	Log-Likelihood Ratio
LNC	Lattice Network Coding
LSD	Layered Soft Detector
MA	Multiple Access
MAC	Multiple Access Channel
MAP	Maximum A-posteriori Probability
MED	Minimum Euclidean Distance
MIMO	Multiple Input, Multiple Output
ML	Maximum Likelihood
MLAN	Modulo Lattice Additive Noise
MLNC	Multilevel Lattice Network Coding
MMSE	Minimum Mean Square Error
MSD	Multi-Stage Decoder
MSE	Mean Square Error
NC	No Correlation
NCM	Network Coded Modulation
NTF	Network Transfer Function
pdf	probability density function
PEP	Pairwise Error Probability
PI-C	Perfect Interference Cancellation
PID	Principal Ideal Domain
PHY	Physical Layer
QAM	Quadrature Amplitude Modulation
Rx	Receiver
SC	Superposition Coding
SCFD	Successive CaF Decoding
SD	Successive Decoding
SD-HIC	Successive Decoding with Hierarchical Interference Cancellation
SD-IC	Successive Decoding with Interference Cancellation
SDR	Software-Defined Radio
SER	Symbol Error Rate

SINR	Signal-to-Noise-and-Interference Ratio
SISO	Soft Input Soft Output
SLS	System-Level Simulator
SMN	Smart Meter Network
SNF	Smith Normal Form
SNR	Signal-to-Noise Ratio
SODEM	Soft-Output Demodulator
TDMA	Time Division Multiple Access
TE	Technical Element
Tx	Transmitter
UDP	User Datagram Protocol
USRP	Universal Software Radio Peripherals
VN	Variable Node(s)
WBN	Wireless Butterfly Network
WCN	Wireless Cloud Network
WP	Work Package
WPLNC	Wireless Physical Layer Network Coding
XOR	eXclusive OR

Part I

Introduction

1 Introduction

1.1 Goals and context of the report

This WP4 deliverable includes all techniques related to terminal node processing, including the Wireless Physical Layer Network Coding (WPLNC). It introduces all functionalities related to the output signal generation, e.g. modulation or channel coding, and to the processing of the received signal at the particular nodes' input(s).

The main goal of this WP4 deliverable is two-fold:

- It provides a comprehensive technical overview of selected physical layer techniques and algorithms which has been successfully implemented by the HW and SLS demonstrators (as reported in the WP5 deliverables).
- It summarises additional promising physical layer techniques and algorithms developed in the last stage of the project.

1.2 Background and related work

1.2.1 Wireless physical layer network coding

A major part of the work in this report is based on the WPLNC principles. A comprehensive overview od WPLNC and related theory is beyond the scope of this report. The structured list of references related to the WPLNC technique is summarised below:

- Layered design of Network Coded Modulation (NCM) and fundamental limits for Hierarchical Decode & Forward (HDF) Multiple-Access (MAC) phase [1, 2],
- HDF Broadcast (BC) phase with partial hierarchical side-information [3–5],
- Parametric NCM design [6–8],
- Denoising and optimised symbol decision maps [9–11],
- Relaying PHY technique [12, 13],
- Lattice based NCM design [14–19],
- General [20, 21].

1.2.2 Compute and forward

In addition to a general WPLNC, several contributions in this report refer to a Compute and Forward (CaF) [22] strategy, which is a form of WPLNC based on the nested lattice codes. The lattices codes can be based on different types of underlying code. In this case the hierarchical network code mapping functions applied at each relay rely on integer coefficients chosen at each relay. CaF can be considered as a good candidate for dense wireless networks such as the DIWINE cloud.

1.3 Structure of this report

The main technical content of this report is presented in two separate parts:

Part II: 'Network coded modulation' presents techniques for modulation and coding design for partial hierarchical side information (Chapter 2), Compute and Forward (Chapter 3), design of Hierarchical Network Code (HNC) maps for large scale scenarios (Chapter 4) and Apmlify and Forward-based NCM design (Chapter 5).

Part III: 'Signal processing and decoding strategies' covers the topics related to signal processing and relay/destination decoding operations, including the processing of hierarchical interference (Chapter 6), analysis of the pairwise-error probability of hierarchical decoders (Chapter 7) and practical implementation of compute and forward techniques (Chapter 8)

The particular WP4 techniques/algorithms (denoted as Technical Elements (TE) in this report) developed in the DIWINE project are presented separately in (relatively standalone) sections. We believe that this approach can help the reader to get easily acquainted with a desired technique/algorithm, without the necessity to delve into the details of the complete report.

In the rest of this section we provide an overview of all TE implemented by the DIWINE demonstrators (both HW and SLS) and we summarise the additional TE, which were developed in the last stage of the project, and thus were not considered for demonstrator.

1.3.1 DIWINE technical elements implemented by demonstrators

Coded superposition modulation presented in sections 2.1, 2.2 provides a systematic design tool for a practical *adaptive coded constellation* for relaying in a 5-node, single-relay butterfly network. The proposed technique allows to match the constellation and channel encoder parameters to the actual channel conditions, and thus it provides a reliable practical communication scheme for various channel conditions. The SMN HW demonstrator implementation of this TE is reported in [D5.42].

Cloud access node scheduling techniques treated in Section 4.3 interprets the cloud as a macro-relay network with Cloud Access (CA) nodes as gateways where packets are queued and flooded adaptively according to the queue state toward the macro-relay. The traffic and interference within the macro-relay is modelled using an approximation for parallel and mutually interacting flows that is appropriate to be validated within the SLS. The most efficient scheduler that complies with delay sensitive applications (such as in CIMC applications) when traffic is heterogeneous needs to offload the packets among CA nodes. The SLS demonstration of this TE appeared in [23].

1.3.2 Additional DIWINE technical elements

Non-cooperative compute-and-forward strategies treated in Section 3.1 deal with outage probability caused by local optimisation of computation rate at each relay. We propose new non-cooperative techniques on how to choose the desired equations and enforce linear independent equations at the destination. This allows to solve the optimisation problem in a localised and distributed manner.

Multilevel lattice network code design framework based on an algebraic construction called Elementary Divisor Construction (EDC) which subsumes commonly-used Constructions A and D is described in Section 3.2. It enables a multilevel lattice construction which allows higher rate lattice codes (and hence also CaF schemes) to be implemented without using higher order codes, which would require much more complex decoders. Alternative decoding algorithms are also described.

Network transfer function and stage scheduling are developed in Sections 4.1 and 4.2. In order to properly design the NCM transmitted by network nodes and the relay processing, a formal description of the global network processing function is required. Any practical implementation also implies the half-duplex constrained relays which imposes the network to work in multiple stages. A polynomial based formalism defining the Hierarchical Network Transfer Function (H-NTF) captures all phenomena related to the stage dependent transmit and receive activity over the network. A half-duplex stage scheduling algorithm is developed using the polynomial formalism of H-NTF.

n-largest eigenmode relaying techniques are introduced in Section 5.1 to guarantee capacity-achieving transmission in the relay nodes. The performance of a dual-hop amplify-and-forward MIMO relay network wherein the relay node has access to partial channel knowledge is studied. It is pointed out that relays with multiple antennas are an essential part of advanced communication systems, in particular, dense networks such as the DIWINE cloud and that due to the dense nature of the network, assuming perfect channel state information is unrealistic in practice.

Successive decoding in WPLNC systems presented in Section 6.1 discusses the problem of interference cancellation in WPLNC systems, where only a function of user data can be decoded instead of separate user data. We show that that even in this case the knowledge of hierarchical data can be efficiently exploited in the decoding process to significantly reduce the impact of interfering signal on the subsequent decoding operations in WPLNC systems.

Hierarchical interference cancellation using successive CaF decoding is addressed in Section 6.2. This technique allows increasing the number of degrees of freedom in lattice misalignment equaliser while using all available hierarchical (many-to-one function) auxiliary equations. This technique is not constrained to use only integer linear map combinations and allows more freedom in choosing a given desired codeword map (equation) at the relay in a complicated multi-stage network.

Hierarchical pairwise error probability reported in Section 7.1 provides an essential tool allowing to design practical optimised coding schemes. It reveals the connection between the performance and the decoding metric that is directly related to the codeword and/or constellation properties. Unlike for the single-user case, the hierarchical pairwise error probability reveals a complicated dependence on the structure of the hierarchical codeword/constellation.

Complex low density lattice codes and their application to lattice network coding is described in Section 8.1. Complex low density lattice codes, unlike other lattices, are designed directly in the complex Euclidean space, and defined by a generator matrix whose inverse in sparse. It may therefore be decoded using a message-passing algorithm, similar to the decoding algorithm for LPPC codes. The section describes practical codes and decoding algorithms, and gives simulation results for example codes.

Convolutional lattice encoding and decoding is investigated in Section 8.2, emphasising on constructing lattices from convolutional codes based on Construction A and D and analysis of the error performance. We show that practical implementation of physical layer network coding using CaF in dense relay networks requires practical lattice codes with reasonable decoding complexity. The focus is on lattice decoding methods that exploit the trellis structure of the lattice, ensuring affordable complexity in dense relay networks.

Part II

Network coded modulation

2 Layered NCM for partial HSI scenarios

ï»¿

2.1 Superposition constellation design

The WPLNC techniques exploit the inherent nature of wireless channels to improve the performance of wireless communication systems. Unlike conventional network coding, in WPLNC the transmitted signals constructively interact directly in the constellation space, thus inducing specific requirements on the source constellation design. Suitable multi-source constellations should enable *direct decoding of WPLNC functions* of user data (from the observed superimposed constellations at receiving nodes) and simultaneously they should allow *delivery of partial information* to nodes with worse channel conditions (exploiting the natural broadcast property of wireless channels). Source constellations possessing both the aforementioned attributes *simultaneously* can be desirable in a WPLNC-based system, and thus proper *constellation design* can become a relatively challenging task. In this report we focus on this problem and we introduce a systematic *constellation design algorithm* for a 5-node *Wireless Butterfly Network* (WBN) with WPLNC processing, where the basic principles of *multi-source constellation design for WPLNC systems* can be demonstrated. We show that the proposed constellations outperform the conventional approaches over the whole range of SNR conditions in the system.

2.1.1 Introduction

The invention of WPLNC techniques has indeed provided a means for a significant enhancement of wireless system performance [21, 24]. However, WPLNC processing has also revealed several non-trivial research problems which do not have their counterparts in conventional point-to-point or multi-user systems, like the sensitivity to channel parameterisation [8, 9, 25, 26] or challenging multi-source transmission synchronisation [27–29]. Furthermore, the specific characteristics of WPLNC systems also call for a *novel constellation design.*

WPLNC relaying techniques exploit the inherent superposition nature of wireless channels to enable *direct decoding* of *specific WPLNC functions of user data* from the received (superimposed) signals [1, 30]. In addition WPLNC aims to exploit the specific *broadcast property* of wireless channels [24] to reduce the consumption of channel resources.

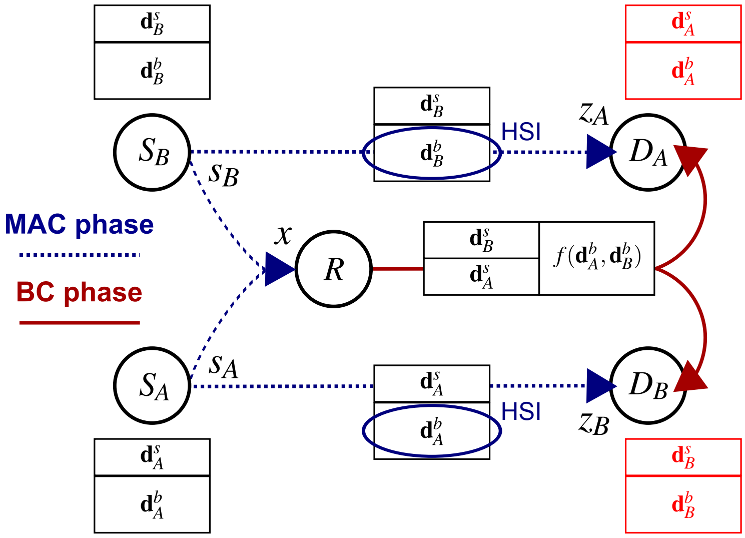


Figure 2.1: Symmetric WBN model with half-duplex constraint. In the first step (Multiple Access – MA) the relay R receives a noisy superimposed constellation $x = h_{AR}s_A + h_{BR}s_B + w_R$, while destination D_A (respectively D_B) overhears $z_A = h_{BA}s_B + w_A$ (respectively $z_B = h_{AB}s_A + w_B$) from the source S_B (respectively S_A). Scalar complex channel coefficients h_{ij}, $i, j \in \{A, B, R\}$, $i \neq j$ are assumed to be constant during the communication round and known at the respective receiving node. Zero mean i.i.d. complex Gaussian noise sample $w_i, i \in \{A, B, R\}$ has variance σ_i^2. Source constellations have unit energy per symbol and consequently the source→relay MAC channel $((S_A,\ S_B) \rightarrow R)$ has SNR $\gamma_{\text{MAC}} = {}^{|h_{AR}|^2}/_{\sigma_R^2} = {}^{|h_{BR}|^2}/_{\sigma_R^2}$, the unintended source→destination (HSI) channels ($S_A \rightarrow D_B$, $S_B \rightarrow D_A$) have SNR $\gamma_{\text{HSI}} = {}^{|h_{BA}|^2}/_{\sigma_A^2} = {}^{|h_{AB}|^2}/_{\sigma_B^2}$ and the relay output broadcast channel (BC) has SNR $\gamma_{\text{BC}} = {}^{|h_{RA}|^2}/_{\sigma_R^2} = {}^{|h_{RB}|^2}/_{\sigma_R^2}$.

Source constellations possessing both the aforementioned attributes[1] *simultaneously* can be desirable in a WPLNC-based system and thus a proper *constellation design* can become a relatively challenging task.

In this report we focus on the constellation design problem in the 5-node *Wireless Butterfly Network* (WBN), see Figure 2.1, where the basic principles of *multi-source constellation design for WPLNC systems* can be demonstrated. We provide the following results and contributions:

1. We introduce a *systematic constellation design algorithm*, capable of producing a suitable constellation for arbitrary channel conditions in WBN.

[1] The ability to compose a *suitable superimposed constellation* at a receiving node (allowing direct decoding of specific functions of user data) and the ability to respect the *broadcast nature* of wireless channels (allowing delivery of partial information to nodes with worse channel conditions).

2. We analyse the performance of the proposed constellations and we compare it with the state-of-the-art constellation design.

3. We provide *SNR mapping regions*, defining the optimal (in the sense of overall throughput) constellation design parameters for adaptive WBN relaying.

4. We compare the numerical performance results with a *real-world HW* evaluation.

2.1.2 WPLNC relaying in wireless butterfly networks

We assume a *symmetric* WBN model [4,31] with a half-duplex constraint (see Figure 2.1). Since the direct channels between the intended source→destination pairs are not available, the help of the intermediate relay node is necessary to enable end-to-end communication. In this section we focus on the constellation design for an uncoded system[2]. The signal space representations of the transmitted channel symbols in the uncoded system are directly $s_A = \mathcal{A}_s^A(\mathbf{d}_A)$, $s_B = \mathcal{A}_s^B(\mathbf{d}_B)$, where $\mathcal{A}_s^i(\cdot)$ is the memoryless constellation mapper and $\mathbf{d}_i$ is the source $i \in \{A, B\}$ data symbol.

Each communication round can be divided into two steps. In the first step both sources S_A, S_B simultaneously transmit their signal to the relay, while their transmission is overheard by the "unintended" destinations ($S_A \to D_B$, $S_B \to D_A$) due to the broadcast nature of wireless channels. Even though the overheard information does not carry the desired data for the respective destination, it can be efficiently exploited to enable WPLNC-based processing in WBN [4]. Since the WPLNC-encoded signal is sometimes denoted as the *hierarchical signal* [1], we denote this overheard "complementary" information as the *Hierarchical Side Information* (HSI) [31].

The relay performs WPLNC-based decoding [1] of the received signal and broadcasts the decoded WPLNC (hierarchical) information as s_R to D_A, D_B in the second step. Both destinations are able to decode the desired data using the relay's (hierarchical) signal and the HSI that it overhears, if proper WPLNC processing is employed in the system.

A suitable information-theoretic relaying strategy (based on the Superposition Coding (SC) [32,33]) for WBN was proposed in [4,31]. The fundamental idea of the SC approach is to split each data symbol $\mathbf{d}_i$ at source $i \in \{A, B\}$ into the *basic* $\mathbf{d}_i^b = [d_{i,0}^b, \ldots, d_{i,N_b-1}^b]$ and *superposed* $\mathbf{d}_i^s = [d_{i,0}^s, \ldots, d_{i,N_s-1}^s]$ parts ($d_{i,n}^k \in \{0, 1\}$, $k \in \{b, s\}$) and process each of the resulting data streams separately throughout the system. A simplified description of the SC-based relaying in an uncoded WBN system is summarised in Table 2.1 and depicted in Figure 2.1.

[2] In Section 2.2 we show that an arbitrary binary error protection code can be readily used with the proposed constellations in a practical system.

Step 1: Source transmission

- Source S_i, $i \in \{A, B\}$ maps its data symbol $\mathbf{d}_i = \left[\mathbf{d}_i^b; \mathbf{d}_i^s\right]$ into a $(N_b + N_s)$-bit superposition modulation symbol

$$s_i = \mathcal{A}_s^i\left(\left[\mathbf{d}_i^b; \mathbf{d}_i^s\right]\right), \tag{2.1}$$

- S_A, S_B simultaneously transmit their signals to the relay and unintended destinations (Figure 2.1).

- Relay receives

$$x = h_{AR}\mathcal{A}_s^A\left(\left[\mathbf{d}_A^b; \mathbf{d}_A^s\right]\right) + h_{BR}\mathcal{A}_s^B\left(\left[\mathbf{d}_B^b; \mathbf{d}_B^s\right]\right) + w_R \tag{2.2}$$

  and decodes $\left[\mathbf{d}_A^s; \mathbf{d}_B^s; f(\mathbf{d}_A^b, \mathbf{d}_B^b)\right]$, where f is a hierarchical WPLNC function [1].

- D_j receives

$$z_j = h_{ij}\mathcal{A}_s^i\left(\left[\mathbf{d}_i^b; \mathbf{d}_i^s\right]\right) + w_j, \tag{2.3}$$

  where i, $j \in \{A, B\}$, $i \neq j$ and stores the signal for further processing.

Step 2: Relay broadcast

- Relay sends $2N_s + N_b$-bit modulation symbol to both destinations:

$$s_R = \mathcal{A}_s^R\left(\left[\mathbf{d}_A^s; \mathbf{d}_B^s; f(\mathbf{d}_A^b, \mathbf{d}_B^b)\right]\right). \tag{2.4}$$

- D_j decodes:
  - $\left[\mathbf{d}_A^s; \mathbf{d}_B^s; f(\mathbf{d}_A^b, \mathbf{d}_B^b)\right]$ (from the relay signal (2.4))
  - $\mathbf{d}_i^b$ (from the stored signal z_j (2.3), after interference cancellation of known $\mathbf{d}_i^s$)
  - $\mathbf{d}_j^b$ (from $\mathbf{d}_i^b$ and $f(\mathbf{d}_A^b, \mathbf{d}_B^b)$, using a standard WPLNC decoding [1])

- D_j merges $\mathbf{d}_j^b$ with $\mathbf{d}_j^s$ to obtain $\left[\mathbf{d}_j^b; \mathbf{d}_j^s\right]$.

Table 2.1: SC relaying scheme in uncoded WBN.

2.1.3 Superposition modulation design

A direct implementation of a practical modulation-coding scheme based on the SC approach is unfortunately non-trivial. The simplest approach is to firstly design a suitable constellation mapper $\mathcal{A}_s^i$ (including proper bit-mapping) which produces the output constellation symbols $s_i = \mathcal{A}_s^i\left(\left[\mathbf{d}_i^b; \mathbf{d}_i^s\right]\right)$, $s_i \in \mathbb{C}^1$ and then append individual error correction encoders separately for basic and superposed data streams.

Since an arbitrary error protection code suitable for single-user channels can be readily used in this case, we focus on the design of the constellation mappers $\mathcal{A}_s^A$, $\mathcal{A}_s^B$. We follow the basic idea of *Superposition Modulation* (see e.g. [34] and references therein) to produce the output constellation as

$$s = \mathcal{A}_s(\left[\mathbf{d}^b; \mathbf{d}^s\right]) = \sum_{n=0}^{N_s-1} L_n^s(2d_n^s - 1) + \sum_{n=0}^{N_b-1} L_n^b(2d_n^b - 1), \tag{2.5}$$

where L_n^b, L_n^s are the *scaling coefficients.*

Generally in superposition modulation [34], the scaling coefficients L_n^b, L_n^s (2.5) are taken from the set of complex numbers, which allow the joint optimisation of both the power level and signal space angle of the resulting constellation points. In this report we follow a slightly simplified approach, and we choose the scaling coefficients L_n^b, L_n^s from a set of purely imaginary or real numbers.

As shown in Table 2.1, the basic and superposed parts of the source information are processed in a different way through the system, and hence we will discuss the constellation design for these two specific information streams separately. We briefly introduce the incentives and main ideas of the design for the superposed and basic parts, and then we summarise the complete multi-source constellation design in a systematic algorithm.

Constellation design: Superposed part

The superposed information given by $\mathbf{d}_A^s$, $\mathbf{d}_B^s$ has to be jointly decoded by the relay (Table 2.1), and hence a source constellation avoiding all potential overlaps in the superimposed constellation (observed by the relay node) is desirable. In an AWGN channel ($h_{AR} = h_{BR} = 1$ in Figure 2.1) this can be simply achieved by mapping the information bits to the real-valued ASK constellation at S_A and purely imaginary-valued ASK constellation at S_B (or vice versa). The resulting superimposed constellation at the relay will then form a regular QAM superimposed constellation without any overlapping of constellation symbols. This can be easily achieved by setting the coefficients[3] as $L_{A,n}^s = 2^n$ and $L_{B,n}^s = \mathrm{j} \cdot 2^n$.

[3] A smaller average number of nearest neighbours in the resulting constellation at the unintended destination can be achieved by altering the real and imaginary domains between the consecutive scaling coefficients at S_A, S_B.

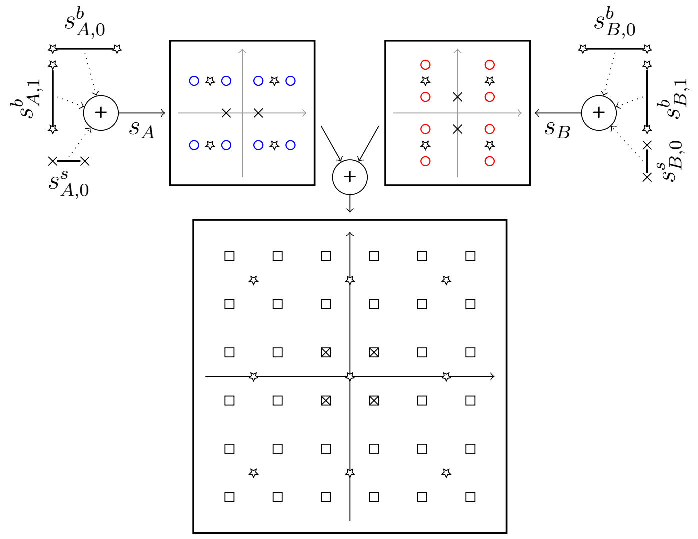


Figure 2.2: Source constellation design example for $N_b = 2$, $N_s = 1$. Resulting constellations are depicted as blue circles (S_A output constellation), red circles (S_B output constellation) and squares (received superimposed constellation at R). Hierarchical function is $f\left(\mathbf{d}^b_A,\, \mathbf{d}^b_B\right) = \mathbf{d}^b_A \oplus \mathbf{d}^b_B$.

Constellation design: Basic part

Unlike the superposed part, only a specific WPLNC function of the basic information $\mathbf{d}^b_A$, $\mathbf{d}^b_B$ has to be decoded by the relay (Table 2.1), and hence some overlapping of symbols in the superimposed constellation (observed by the relay node) can be allowed[4]. More precisely, two (or more) pairs of basic information symbols $\left(\mathbf{d}^b_A,\, \mathbf{d}^b_B\right)$, $\left(\mathbf{d}'^b_A,\, \mathbf{d}'^b_B\right)$ can be allowed to fall into the same point in the superimposed constellation:

$$s_A\left(\mathbf{d}^b_A\right) + s_B\left(\mathbf{d}^b_B\right) = s_A\left(\mathbf{d}'^b_A\right) + s_B\left(\mathbf{d}'^b_B\right), \tag{2.6}$$

iff they belong to the same WPLNC function output, i.e. iff $f(\mathbf{d}^b_A, \mathbf{d}^b_B) = f(\mathbf{d}'^b_A, \mathbf{d}'^b_B)$. In this case the overlapping symbols produce the same hierarchical output, which can be successfully resolved by the destination, if a matching HSI information is available (Table 2.1).

The particular choice of the scaling coefficients $L^b_{A,n}$, $L^b_{B,n}$ depends on the particular choice of the hierarchical WPLNC function. In the following subsections we discuss the modulation design for two common choices of WPLNC function, namely the bit-wise XOR [21] and Modulo-sum [35] operations.

[4] This in turn results in a superimposed constellation with improved Euclidean distance properties.

Bitwise-XOR hierarchical function. Suitable overlapping of constellation symbols at the relay is provided by letting both sources transmit at the same level, i.e. $L^b_{A,n} = L^b_{B,n}$. It can be easily shown that if these coefficients are chosen as $L^b_{A,n} = L^b_{B,n} = 3^{\lfloor n/2 \rfloor}$ (for even n) and $L^b_{A,n} = L^b_{B,n} = \mathrm{j}\, 3^{\lfloor n/2 \rfloor}$ (for odd n), the received superimposed constellation at the relay will have overlaps only among the constellation symbols $\left(\mathbf{d}^b_A,\, \mathbf{d}^b_B\right)$, $\left(\mathbf{d'}^b_A,\, \mathbf{d'}^b_B\right)$ which have the same hierarchical WPLNC output i.e.

$$f(\mathbf{d}^b_A, \mathbf{d}^b_B) = f(\mathbf{d'}^b_A, \mathbf{d'}^b_B) = \mathbf{d}^b_A \oplus \mathbf{d}^b_B,$$

where $\oplus$ is the bit-wise exclusive OR operation.

Modulo-sum hierarchical function. The specific constellation indexing proposed in [35] enables decoding of the Modulo-sum[5] hierarchical function from the superimposed constellation observed by the relay. This allows us to choose the coefficients as $L^b_{A,n} = L^b_{B,n} = 2^{\lfloor n/2 \rfloor}$ (for even n) and $L^b_{A,n} = L^b_{B,n} = \mathrm{j}\, 2^{\lfloor n/2 \rfloor}$ (for odd n), which would not be possible for the bitwise-XOR hierarchical function[6].

Systematic constellation design algorithm

The constellation scaling principles for the basic and superposed parts can be combined, if the scaling coefficients of basic information streams (L^b_n) are pre-scaled by a factor 2^{N_s} to ensure that they lie above the largest scaling coefficient of the superposed part. This in turn allows to design a *multi-source superposition modulation* (possessing both desired properties) for arbitrary N_b, N_s, or, equivalently, to design the constellation mappers $\mathcal{A}^A_s(\cdot)$, $\mathcal{A}^B_s(\cdot)$ for arbitrary quality of HSI channels in WBN. The proposed systematic constellation design is summarised in Algorithm 1. An example design of source constellations for $N_b = 2,\ N_s = 1$ is presented in Figure 2.2.

2.1.4 Performance analysis

For the sake of simplicity of the theoretical performance analysis, we assume that all channels in WBN are AWGN, i.e. $h_{ij} = 1$ for $i, j \in \{A, B, R\}$, $i \neq j$. We will evaluate the throughput given by $(N_b + N_s) \cdot (1 - P_e^{\mathrm{FER}})$, where P_e^{FER} denotes the frame error rate. Due to the system symmetry we evaluate the throughput of $S_A \to D_A$ only. We set the frame length to $M = 768$ symbols and the relay output mapper $\mathcal{A}^R_S$ will be defined as a conventional $(2^{N_b + 2N_s})$-QAM constellation.

[5] For the purpose of this report, the Modulo-sum function is defined as $f\left(\mathbf{d}^b_A,\, \mathbf{d}^b_B\right) = \mathcal{D}^{-1}\left(\left[\mathcal{D}\left(\mathbf{d}^b_A\right) + \mathcal{D}\left(\mathbf{d}^b_B\right)\right] \mod 2^{N_b}\right)$, where $\mathcal{D}(\mathbf{d})$ represents a decimal integer representation of a binary vector $\mathbf{d}$ and $\mathcal{D}^{-1}(m)$ represents a binary vector representation of a decimal integer m.

[6] We will show that the system with Modulo-sum hierarchical function potentially outperforms the system with bit-wise XOR function. However, unlike the bit-wise XOR, implementation of error correction coding could become quite challenging for Modulo-sum mapping at the relay.

Algorithm 1 Constellation design for fixed $N_b,\ N_s$.

```
**Input:** $(N_b,\ N_s)$
**for** $n \in \{0, \ldots, N_s - 1\}$                                  ▷ "superposed levels"
  $L^s_{A,n} \leftarrow 2^n$, $L^s_{B,n} \leftarrow \mathrm{j}\,2^n$, for even $n$
  $L^s_{A,n} \leftarrow \mathrm{j}\,2^n$, $L^s_{B,n} \leftarrow 2^n$, for odd $n$
**end**
**for** $n \in \{0, \ldots, N_b - 1\}$                                  ▷ "basic levels"
  **switch** hierarchical function $f$:
    **case** "bitwise-XOR":
      $L^b_{A,n} = L^b_{B,n} \leftarrow 2^{N_s} \cdot 3^{\lfloor n/2 \rfloor}$ for even $n$
      $L^b_{A,n} = L^b_{B,n} \leftarrow \mathrm{j}\,2^{N_s} \cdot 3^{\lfloor n/2 \rfloor}$ for odd $n$
    **case** "Modulo Sum":
      $L^b_{A,n} = L^b_{B,n} \leftarrow 2^{N_s} \cdot 2^{\lfloor n/2 \rfloor}$ for even $n$
      $L^b_{A,n} = L^b_{B,n} \leftarrow \mathrm{j}\,2^{N_s} \cdot 2^{\lfloor n/2 \rfloor}$ for odd $n$
  **end**
**end**
**for** $i \in \{A, B\}$ and $\forall \mathbf{d}_i = \left[\mathbf{d}^b_i; \mathbf{d}^s_i\right]$, $d^b_{i,n},\ d^s_{i,n} \in \{0, 1\}$:
  **for** $n_1 \in \{0, N_s - 1\},\ n_2 \in \{0, N_b - 1\}$
    $s^s_{i,n_1} = L^s_{i,n_1}\left(2d^s_{i,n_1} - 1\right)$
    $s^b_{i,n_2} = L^b_{i,n_2}\left(2d^b_{i,n_2} - 1\right)$
  **end**
  $\alpha^2 = \sum_{n=0}^{N_s-1} (L^s_{i,n})^2 + \sum_{n=0}^{N_b-1} (L^b_{i,n})^2$              ▷ "norm. coefficient"
  $s_i = \mathcal{A}^i_s\left(\left[\mathbf{d}^b_i; \mathbf{d}^s_i\right]\right) = \alpha^{-1}\left(\sum_{n=0}^{N_s-1} s^s_{i,n} + \sum_{n=0}^{N_b-1} s^b_{i,n}\right)$   ▷ "const. symbols"
**end**
**Output:** Constellation alphabets $\mathcal{A}^A_s,\ \mathcal{A}^B_s$.
```

Throughput analysis

Since all constellations points lie on a regular QAM grid, it is straightforward to evaluate the exact Symbol Error Rate (SER) for all channels analytically. The symbol error probability P_e^{SER} can be approximated[7] by its upper-bound[8] $P_{e,\mathrm{UB}}^{\mathrm{SER}} = 1 - ((1 - P_e^{\mathrm{MAC}})(1 - P_e^{\mathrm{HSI}})(1 - P_e^{\mathrm{BC}}))$, where P_e^{MAC}, P_e^{HSI} and P_e^{BC} are the probabilities of symbol error in the MAC, HSI and BC channels (respectively). Since the system is memoryless, the overall FER is given by $P_e^{\mathrm{FER}} = 1 - (1 - P_e^{\mathrm{SER}})^M$ which can be approximated by $P_{e,\mathrm{UB}}^{\mathrm{FER}} = 1 - (1 - P_{e,\mathrm{UB}}^{\mathrm{SER}})^M$. Nearest neighbour pairwise error approximation can be used for an efficient error rate evaluation.

To compare the performance of the proposed constellations, we evaluate analytically the lower-bound of the throughput as $T_{\mathrm{LB}} = (N_b + N_s) \cdot (1 - P_{e,\mathrm{UB}}^{\mathrm{FER}})$ for fixed γ_{MAC}, γ_{BC} as a function of γ_{HSI}. We analyse all permissible constellations for $N_b + N_s = 2$ (see Figure 2.3) and compare their performance with the throughput T_{ref} of the *reference "conventional" multi-user detection* with regular QAM constellation (see Figures 2.4, 2.5 and 2.8, 2.9). The hierarchical function was set to the bit-wise XOR $f\left(\mathbf{d}_A^b, \mathbf{d}_B^b\right) = \mathbf{d}_A^b \oplus \mathbf{d}_B^b$ in all relevant examples. In addition, the analytically evaluated throughput (lower-bound) we also include the result of a numerical Monte-Carlo evaluation of throughput T_{sim} over a 10^4 frames in all four figures. For $N_b \neq 0$, the lower-bound T_{LB} is relatively tight in all cases, and hence it provides a reasonable approximation of the numerically evaluated throughput T_{sim}.

The reference scenario with 4-QAM fails to provide a non-zero throughput (T_{ref}) in the whole range of analysed SNRs in all four plots[9]. For $(N_b, N_s) = (2, 0)$ the proposed constellations are identical to 4-ary constellations used for conventional WPLNC, see [9]. However, conventional WPLNC requires perfect HSI to enable the decoding of WPLNC functions at destinations, limiting its operation to high γ_{HSI} regions in WBN. On the contrary, the novel proposed constellations provide a *non-zero throughput* even in the *low-to-medium SNR* region of HSI channels. The resulting SNR and throughput gains are highlighted in Figures 2.4, 2.5 and 2.8, 2.9.

SNR mapping regions

As shown in Figures 2.4, 2.5 and 2.8, 2.9 the throughput of the system depends heavily on the actual SNR conditions. This observation calls naturally for an *adaptive constellation design*. The proposed constellation design (Algorithm 1) can be readily used in this case,

[7] Unless stated otherwise, we consider the error probability of the overall single-user transmission chain $S_A \rightarrow D_A$.

[8] The evaluated $P_{e,\mathrm{UB}}^{SER}$ is an upper-bound of P_e^{SER}, since an error in MAC/BC channels does not necessarily induce an error in the overall communication chain. This is especially true if the error occurs in the superposed bits $\mathbf{d}_B^s$ used for interference cancellation processing at D_A.

[9] This is a direct consequence of the fact that the resulting superimposed constellation have too many overlaps, preventing successful multi-user decoding of both of the separate user data streams.

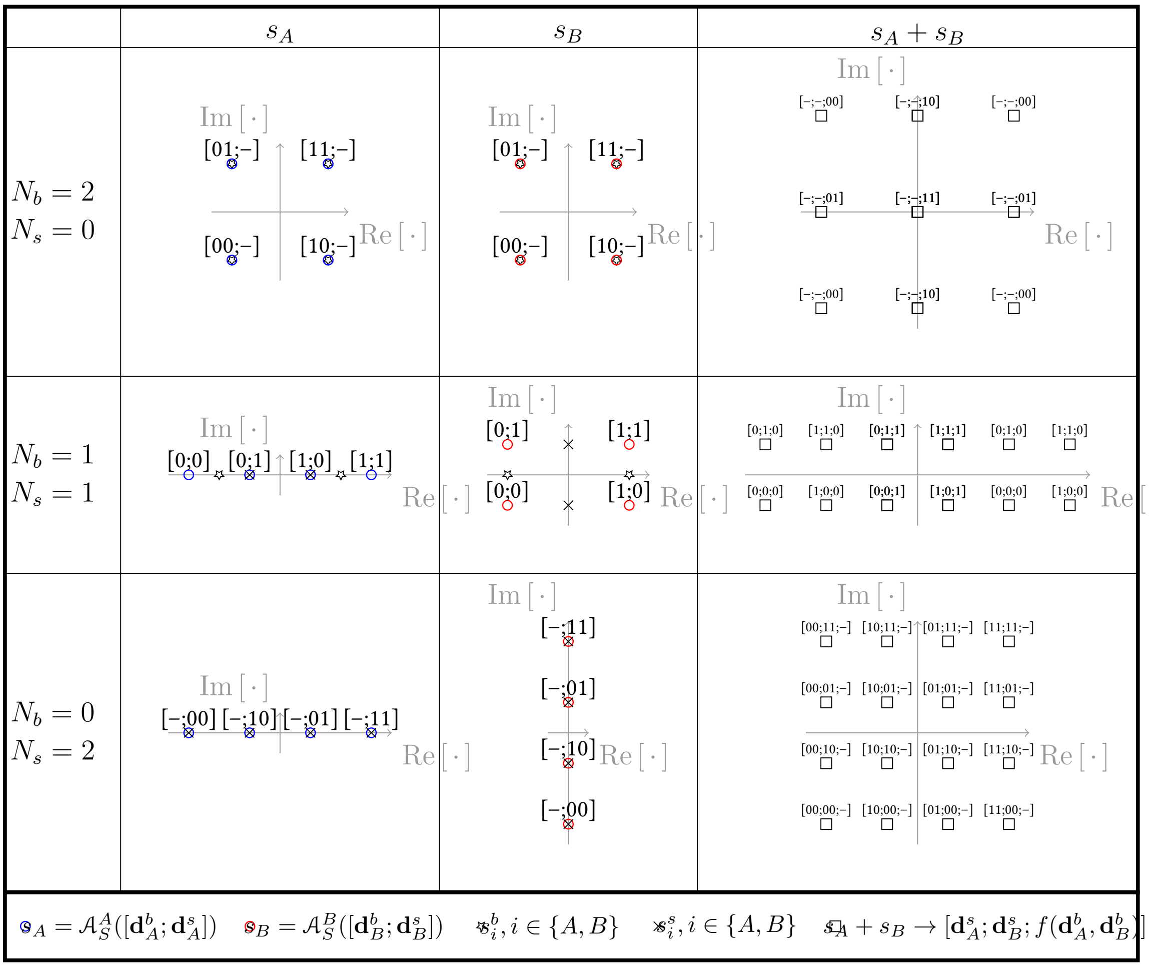


Figure 2.3: Proposed constellation design for $(N_b, N_s) = \{(2,0); (1,1); (0,2)\}$.

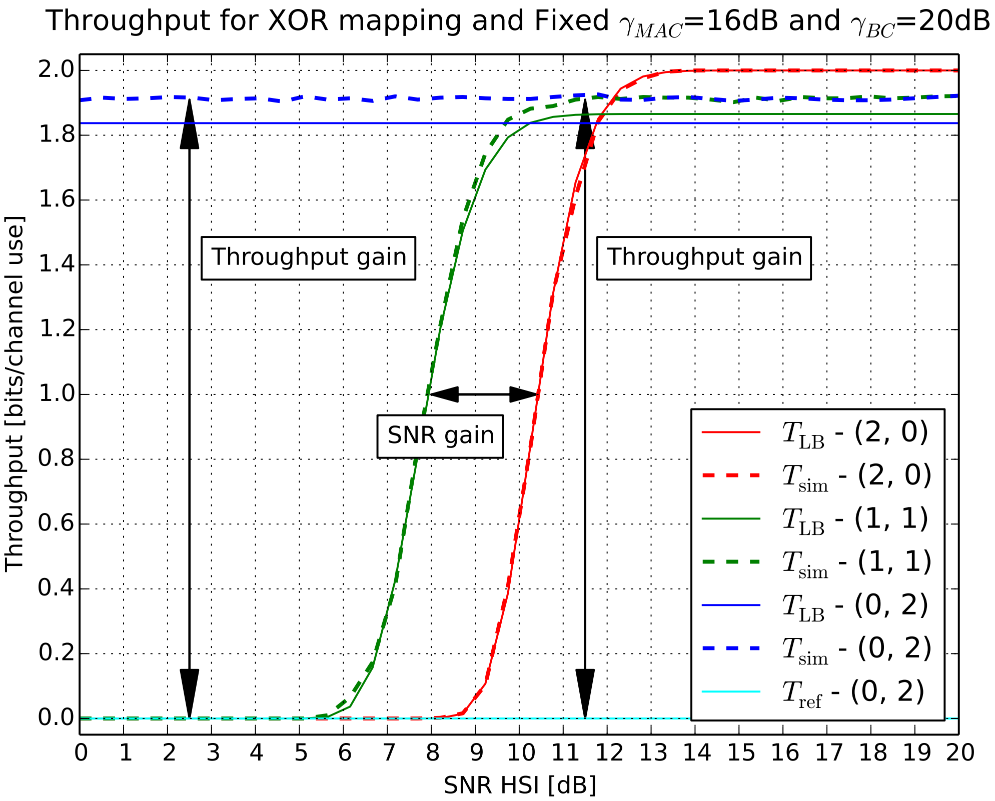


Figure 2.4: Comparison of throughput as a function of γ_{HSI} for $\gamma_{\mathrm{MAC}} = 16\,\mathrm{dB}$, $\gamma_{\mathrm{BC}} = 20\,\mathrm{dB}$ and given (N_b, N_s).

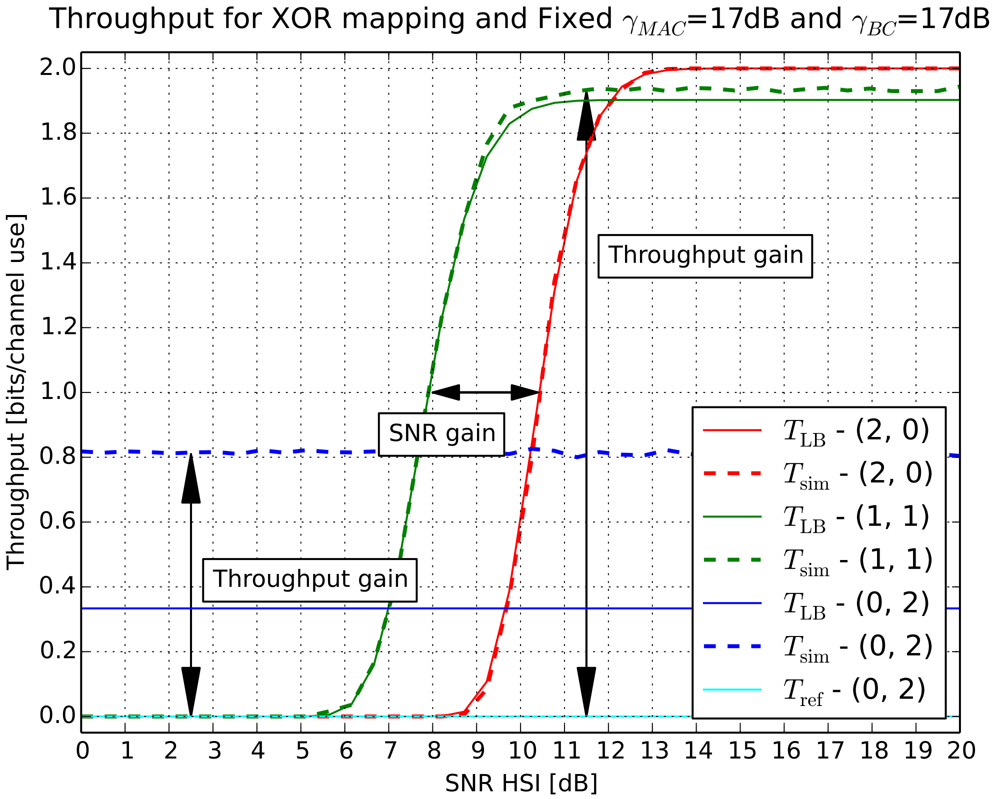


Figure 2.5: Comparison of throughput as a function of γ_{HSI} for $\gamma_{\mathrm{MAC}} = 17\,\mathrm{dB}$, $\gamma_{\mathrm{BC}} = 17\,\mathrm{dB}$ and given (N_b, N_s).

if an optimal (in the sense of maximal throughput) choice of $N_b,\ N_s$ for given SNR conditions is provided. We introduce these *optimal SNR mapping regions* for the case where the relay→destination channels throughput is limited by the maximum number of bits which can be reliably transmitted from the relay node to both destinations[10] ($N_r = 8$ in our case). The optimal SNR mapping regions, including the resulting throughput for the optimal choice of (N_b, N_s), are shown[11] in Figure 2.6 (respectively 2.7) for the bit-wise XOR (respectively Modulo-sum) hierarchical functions. The particular pairs of N_b, N_s achieving the maximum throughput for a given SNR are highlighted in both figures.

2.1.5 Hardware implementation

In addition to the analytical and numerical Monte-Carlo evaluation of throughput, we validate the results of Figures 2.8, 2.9 in a real-world setup. The hardware performance is evaluated using Ettus Research Universal Software Radio Peripherals (USRPs) which are computer-hosted SDRs. For a more detailed description of the HW setup please see the DIWINE deliverable D5.42.

In the USRP results the HSI is passed via UDP such that the HSI SNR can be strictly controlled by adding noise. This also avoids the issue of node visibility where direct links $S_A \to D_A$ and $S_B \to D_B$ exist in the laboratory environment. The throughput evaluated by the HW real world setup T_{USRP} is shown in Figures 2.8, 2.9, including the comparison with the analytical lower-bound T_{LB}, Monte-Carlo evaluation T_{sim} and reference scenario T_{ref}. Extremely close agreement can be seen in all cases.

2.1.6 Discussion, conclusions and future work

We have proposed a *systematic algorithm* for the design of *two-source constellations* in the 5-node WBN. Algorithm 1 is capable of producing a multi-source constellation for HSI channels of arbitrary quality in WBN. We have shown that the proposed constellations outperform the conventional approaches over the whole range of SNR conditions in the system.

The particular *SNR mapping regions*, suggesting the optimal choice of the superposition modulation parameters $N_b,\ N_s$ (Table 2.1) were identified, providing a valuable tool for the *adaptive constellation design* in WBN. Even though we have focused on the pure uncoded constellation design, the proposed constellations can be readily combined with an arbitrary binary encoder, thus providing additional protection against transmission

[10] This in turns imposes a limit on the choice of $N_b,\ N_s$, since N_r has to be greater or equal than $2N_s + N_b$ (see Figure 2.1 or Table 2.1) to enable successful decoding of relay information at both destinations.

[11] An analytically evaluated lower bound (Section 2.1.4) was used for the throughput evaluation in this section due to the numerical complexity of the Monte-Carlo simulations. To further simplify the evaluation, we assume an error-free BC channel $P_e^{BC} = 0$ for $N_r \leq 8$.

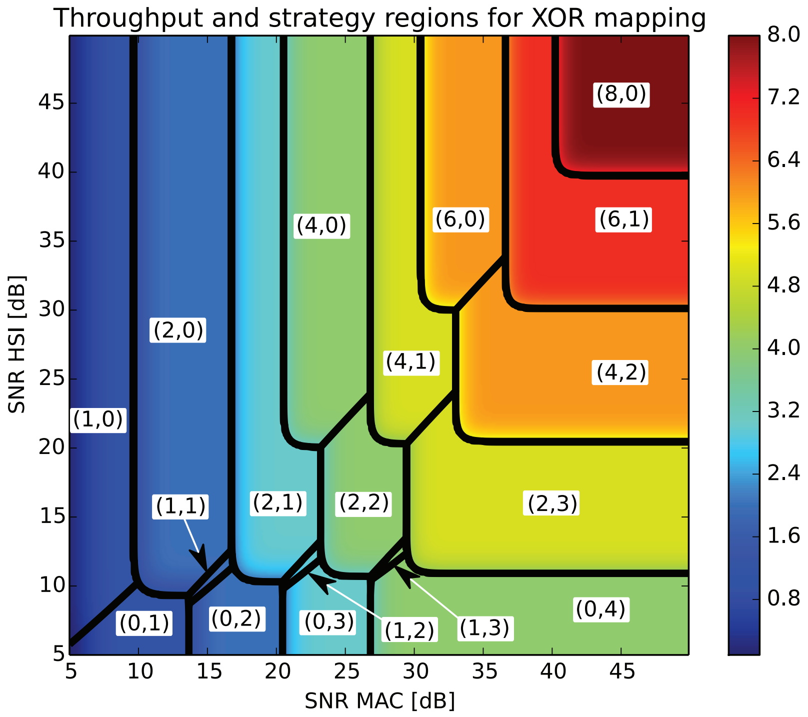


Figure 2.6: Throughput performance T_{LB} and SNR mapping regions: bit-wise XOR hierarchical function.

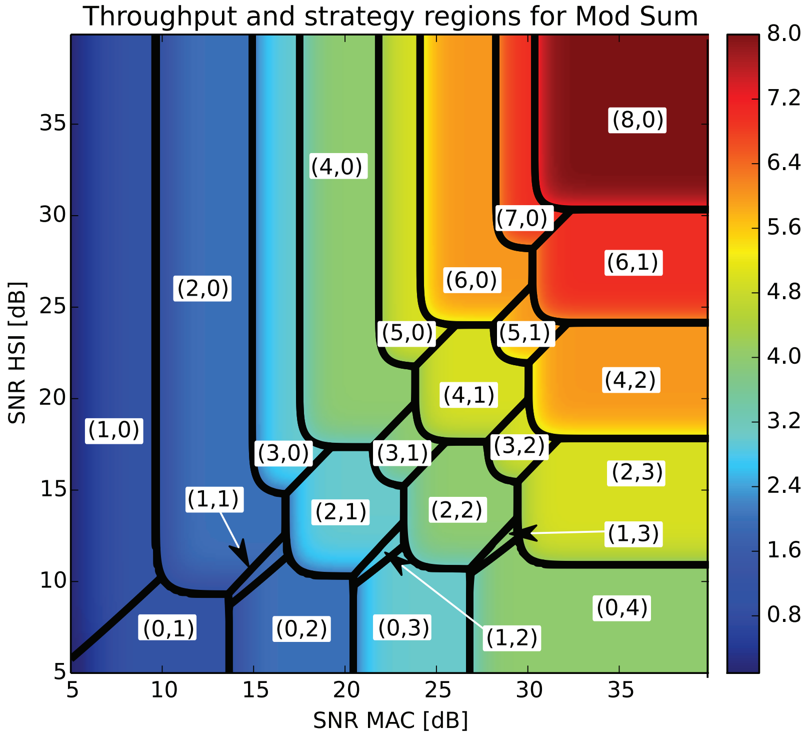


Figure 2.7: Throughput performance T_{LB} and SNR mapping regions: Modulo-sum hierarchical function.

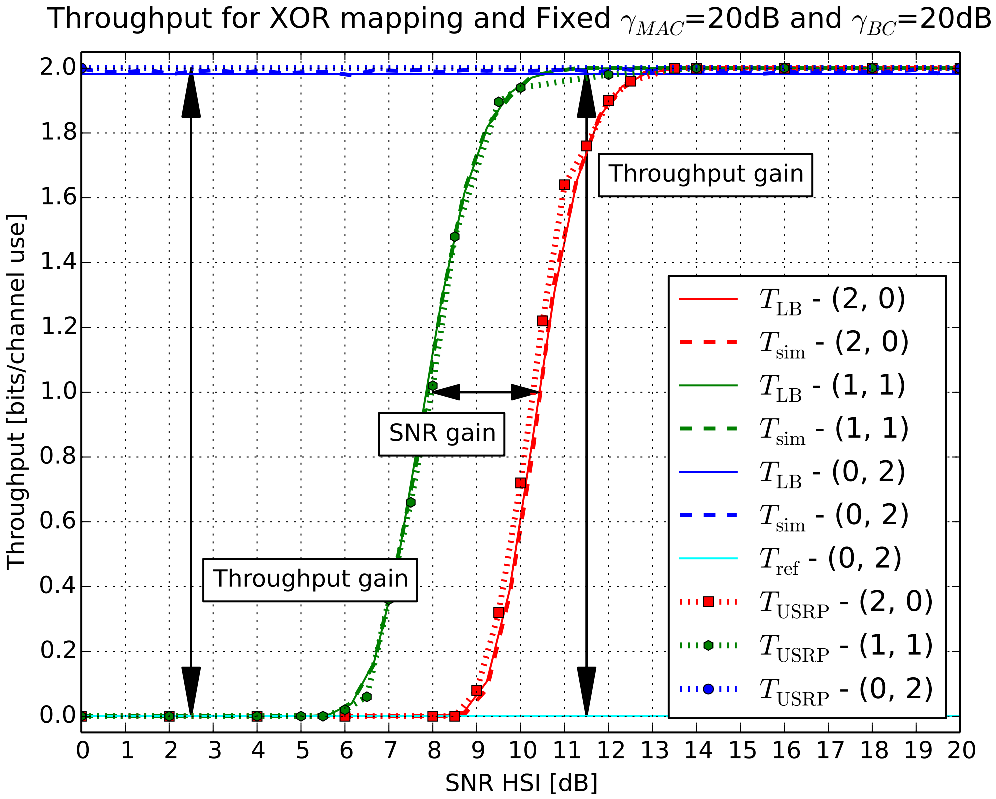


Figure 2.8: Comparison of throughput as a function of γ_{HSI} for $\gamma_{\text{MAC}} = 20\,\text{dB}$, $\gamma_{\text{BC}} = 20\,\text{dB}$ and given (N_b, N_s).

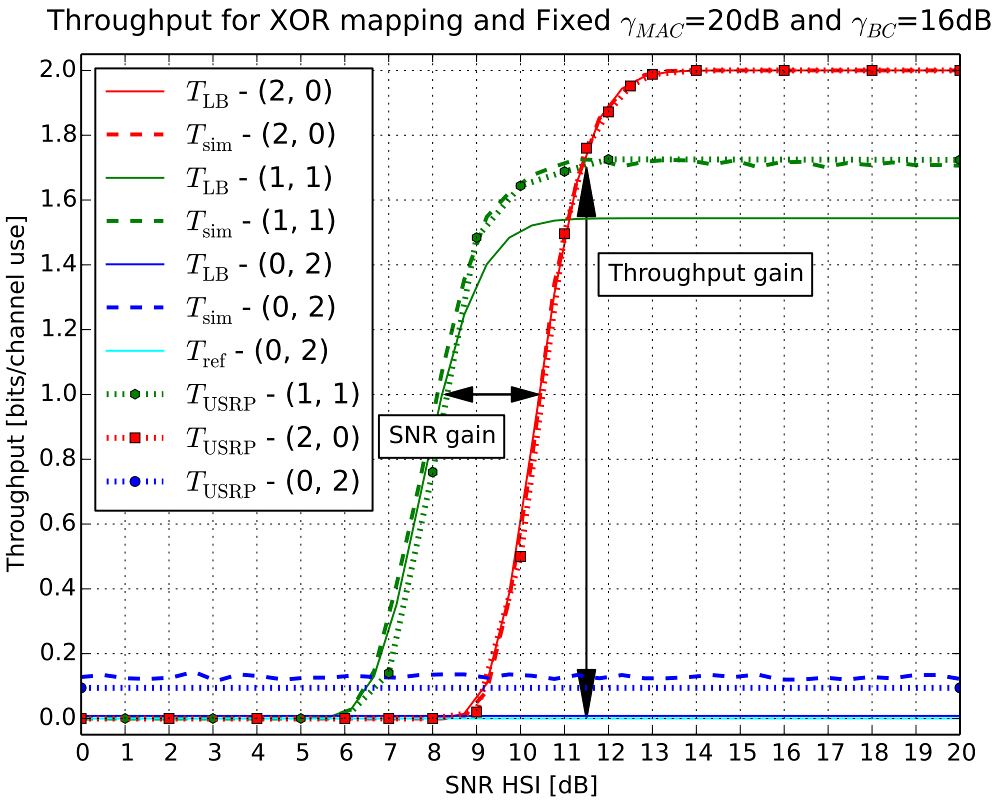


Figure 2.9: Comparison of throughput as a function of γ_{HSI} for $\gamma_{\mathrm{MAC}} = 20\,\mathrm{dB}$, $\gamma_{\mathrm{BC}} = 16\,\mathrm{dB}$ and given (N_b, N_s).

errors. The viability of the proposed constellation design was verified in a real-world HW setup, based on Ettus Research USRPs.

In the next section we show how the error-protection coding can be implemented into the proposed constellation design to provide a reliable adaptive coded modulation processing for practical wireless systems.

2.2 Channel coding extension for the superposition constellation design

In this section we show that forward error correction (channel coding) can be readily appended to the the constellation designed by the Algorithm 1 to provide a reliable physical-layer processing for communication in a real-world environment.

Since it is more straightforward to design a binary channel coding scheme, in this section we discuss the channel coded adaptive modulation scheme for a bit-wise exclusive-or (XOR) WPLNC function ($f(\mathbf{d}_A^b, \mathbf{d}_B^b) = \mathbf{d}_A^b \oplus \mathbf{d}_B^b$, where $\oplus$ is the bit-wise XOR operation).

Note that the presented scheme forms the cornerstone of the SMN HW demonstrator evaluation, as reported in [D5.42].

2.2.1 Introduction

We have shown that Algorithm 1 is a strong tool for the design of source constellations suitable for arbitrary channel SNRs in the WBN system. However, even though the proposed constellations provide promising performance in the uncoded scenario, the extension to a coded system is desirable for practical wireless applications.

Considering the communication in uncoded WBNs (see Section 2.1), there are many ways to improve its reliability through channel coding. In this section we introduce one particular solution, based on the assumption that each source node splits its data into the basic and superposed part and then encodes these parts separately by two constituent channel encoders[12]. We believe that this approach provides the best insight into the channel-coding extension of WBN systems, and hence we focus on its development in the rest of this section.

[12] In a more general approach, each source data word is split into $N_b + N_s$ parallel data sub-words that are encoded separately using the principles of multilevel coding, see [36]. However, the development of a general multilevel coding scheme for WBN is beyond the scope of this report.

2.2.2 Per-link channel coding scheme for symmetric WBN with superposition modulations

In general, it is desirable to protect *all individual transmissions* in the system with error correction coding[13]. Accordingly, the relay node has to decode the desired data (including the particular WPLNC function) from its observation and then it should perform an additional re-encoding in order to protect the subsequent relay→destinations transmission. The resulting channel coding scheme (see Figure 2.10) is introduced in the following text.

Source processing

Source S_A (likewise for S_B) wants to transmit a binary data word $\mathbf{D}_A$ of length $|\mathbf{D}_A| = |\mathbf{D}_B| = k_\mathbf{D}$ to its respective destination D_A. Each data word $\mathbf{D}_A$ is split into the basic and superposed part (similarly as in the uncoded case), i.e.:

$$\mathbf{D}_A = [\mathbf{D}_A^b, \mathbf{D}_A^s] = [d_{A,0}^b, d_{A,1}^b, \dots d_{A,k_\mathbf{D}^b-1}^b, d_{A,0}^s, d_{A,1}^s, \dots d_{A,k_\mathbf{D}^s-1}^s],$$

where $d_{A,n}^b$, $d_{A,n}^s \in \{0,1\}$ and $k_\mathbf{D}^b$, $k_\mathbf{D}^s$ are the lengths of the basic and superposed data sub-words $\mathbf{D}_A^b$, $\mathbf{D}_A^s$ (respectively).

Subsequently, individual $\mathbf{D}_A^b$, $\mathbf{D}_A^s$ are encoded by two separate linear binary encoders $\mathcal{C}_b(\centerdot)$, $\mathcal{C}_s(\centerdot)$ (assumed identical at both sources), producing the constituent codewords as:

$$\mathcal{C}_b(\mathbf{D}_A^b) = \mathbf{C}_A^b = [\mathbf{c}_{A,0}^b, \mathbf{c}_{A,1}^b, \dots, \mathbf{c}_{A,\mathcal{N}-1}^b], \tag{2.7}$$

$$\mathcal{C}_s(\mathbf{D}_A^s) = \mathbf{C}_A^s = [\mathbf{c}_{A,0}^s, \mathbf{c}_{A,1}^s, \dots, \mathbf{c}_{A,\mathcal{N}-1}^s], \tag{2.8}$$

where $\mathcal{N}$ is the number of codeword symbols (equivalently the number of channel uses required to transmit the codeword) and

$$\mathbf{c}_{A,j}^b = [c_{A,j_0}^b, \dots, c_{A,j_{N_b-1}}^b], \tag{2.9}$$

$$\mathbf{c}_{A,j}^s = [c_{A,j_0}^s, \dots, c_{A,j_{N_s-1}}^s] \tag{2.10}$$

are codeword symbols of length N_b (respectively N_s) bits. The dimensionality of binary codewords is thus $\left|\mathbf{C}_A^b\right| = \left|\mathbf{C}_B^b\right| = n_\mathbf{C}^b = \mathcal{N} \cdot N_b$, $|\mathbf{C}_A^s| = |\mathbf{C}_B^s| = n_\mathbf{C}^s = \mathcal{N} \cdot N_s$.

The constituent codewords $\mathbf{C}_A^b$, $\mathbf{C}_A^s$ are then forwarded to the joint constellation mapper $\mathcal{A}^A(\centerdot)$ (provided by Algorithm 1), which produces a sequence of $(N_s + N_b)$-bit constellation symbols $s_{A,j} = \mathcal{A}^A\left(\left[\mathbf{c}_{A,j}^b; \mathbf{c}_{A,j}^s\right]\right)$. Note that in the encoded system it is the codeword

[13] Alternatively, a primitive end-to-end coding scheme can be designed, performing the channel coding solely at sources S_A, S_B and channel decoding (potentially computationally intensive) solely at destinations D_A, D_B, while leaving the relay R to operate on a symbol-by-symbol basis (as in the uncoded system). Unfortunately, this approach is only sub-optimal, as in this case the particular transmissions on source→relay and relay→destination channels are not individually protected by channel coding.

symbols that are mapped to the desired constellation (instead of data symbols), but since $\left|\mathbf{c}_A^b\right| = \left|\mathbf{c}_B^b\right| = N_b$ and $\left|\mathbf{c}_A^s\right| = \left|\mathbf{c}_B^s\right| = N_s$, it is sufficient to formally substitute the codeword symbol $\mathbf{c}_i = \left[\mathbf{c}_i^b; \mathbf{c}_i^s\right]$ for the data symbol $\mathbf{d}_i = \left[\mathbf{d}_i^b; \mathbf{d}_i^s\right]$ in Algorithm 1 (lines 16, 22). The resulting constellation symbols are successively transmitted by S_A (simultaneously with $s_{B,j}$ from S_B) towards the relay node.

Relay processing

The crucial part of the encoded WBN system processing is based at the relay node. Like the uncoded case, the relay has to decode jointly the *superposed* data sub-words ($\mathbf{D}_A^s$, $\mathbf{D}_B^s$) along with the *WPLNC function* of *basic* data sub-words ($f(\mathbf{D}_A^b, \mathbf{D}_B^b) = \mathbf{D}_A^b \oplus \mathbf{D}_B^b$).

Whilst both $\mathbf{D}_A^s$, $\mathbf{D}_B^s$ can be decoded straightforwardly by the relay (using conventional single user decoders $\mathcal{C}_s^{-1}$), the fundamental question is how to decode the WPLNC function $f(\mathbf{D}_A^b, \mathbf{D}_B^b)$ since both $\mathbf{D}_A^b$, $\mathbf{D}_B^b$ are encoded separately at sources S_A, S_B[14]. Fortunately, as proven in [1], the WPLNC function $f(\mathbf{D}_A^b, \mathbf{D}_B^b)$ can be decoded directly from the relay observation, if the particular source data words are encoded by the same linear encoder $\mathcal{C}_b$. The linearity of the code then implies:

$$\begin{aligned} \mathcal{C}_b\left(f(\mathbf{D}_A^b, \mathbf{D}_B^b)\right) &= \mathcal{C}_b\left(\mathbf{D}_A^b \oplus \mathbf{D}_B^b\right) \\ &= \mathcal{C}_b(\mathbf{D}_A^b) \oplus \mathcal{C}_b(\mathbf{D}_B^b) \\ &= \mathbf{C}_A^b \oplus \mathbf{C}_B^b, \end{aligned} \tag{2.11}$$

where $\mathbf{C}_A^b \oplus \mathbf{C}_B^b$ again is a valid codeword, and hence the WPLNC function $f(\mathbf{D}_A^b, \mathbf{D}_B^b)$ can be decoded directly from the relay observation (using a conventional single user decoder $\mathcal{C}_b^{-1}$).

After decoding all required data streams ($\hat{\mathbf{D}}_A^s$, $\hat{\mathbf{D}}_B^s$, $\hat{f}(\mathbf{D}_A^b, \mathbf{D}_B^b)$), the relay creates a joint data word $\mathbf{D}_R = [\hat{\mathbf{D}}_A^s, \hat{\mathbf{D}}_B^s, \hat{f}(\mathbf{D}_A^b, \mathbf{D}_B^b)]$ and re-encodes it as

$$\mathcal{C}_R(\mathbf{D}_R) = \mathbf{C}_R = [\mathbf{c}_{R,0}, \mathbf{c}_{R,1}, \ldots, \mathbf{c}_{R,\mathcal{N}-1}]$$

. The encoded data are then mapped to the output 2^{N_R}-QAM (where $N_R = 2N_s + N_b$) constellation symbols $s_{R,j} = \mathcal{A}^R\left(\mathbf{c}_{r,j}\right)$, $j \in \{0, 1, \ldots, \mathcal{N}-1\}$ and broadcast by the relay to both destination nodes.

[14] Note that separate decoding of $\mathbf{D}_A^b$, $\mathbf{D}_B^b$ and subsequent evaluation of $f(\mathbf{D}_A^b, \mathbf{D}_B^b)$, i.e. conventional joint/multi-user decoding concatenated with traditional network coding [37], can significantly limit the performance of the system [1].

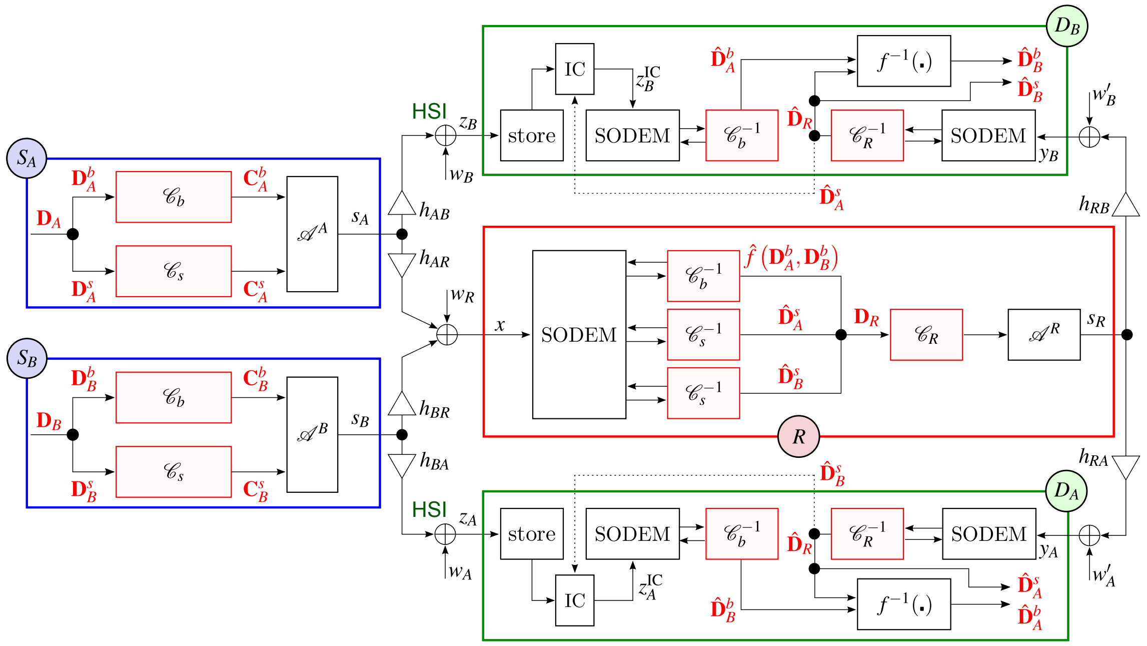


Figure 2.10: Relaying scheme in the encoded WBN system. SODEM stands for a soft-output demodulator, IC is the interference canceller. The channel encoders/decoders which are appended to the uncoded system are emphasised.

Destination processing

Similarly as in the uncoded case, destinations $D_A,\ D_B$ firstly store the signal received in the MA phase, and then, after decoding the relay data word $\mathbf{D}_R$ (from the signal received in the BC phase), both destinations can perform IC to remove the unintended superposed codewords from the MA phase signal to get the desired HSI and then finally recover the desired data (see Figure 2.10 for the details).

2.2.3 Analysis of transmission rates

The last step in the design of a feasible modulation-coding scheme for WBN (see Figure 2.10) is to identify the range of permissible transmission rates for the basic (r_b), superposed (r_s) and relay output (r_R) data streams. To achieve this goal, we analyse the Constellation Constrained (CC) capacities [38] of the proposed source constellations, including the CC capacity of the conventional relay output 2^{N_R}-QAM alphabets (used in the BC phase of communication).

The transmission rates are defined in bits per channel use, and hence $r_i = N_i \cdot {}^{k^i_{\mathrm{D}}}/_{n^i_{\mathrm{C}}}$, where N_i is the number of bits per channel symbol/use in the data stream i, $i \in \{b, s, R\}$. In accordance with the definition of source and relay codewords, we assume that each communication round requires $2\mathcal{N}$ channel uses, where the length of both MA and BC phase is set to $\mathcal{N}$ channel uses. Note that the assumption that BC and MA phases have identical length is generally only suboptimal, see [24,31], but nevertheless it is a common assumption in practical systems, and hence we limit our attention to this particular case.

Relay observation

After splitting the source data streams, the whole system can be interpreted as a 4-user system where the relay has to decode the data $\mathbf{D}_A^s,\ \mathbf{D}_B^s,\ f(\mathbf{D}_A^b, \mathbf{D}_B^b)$ from the *virtual 3-user* ($S_A^s,\ S_B^s,\ S_{AB}^b$) *multiple-access channel* observation (see Figure 2.10). Unfortunately, since one of the users (S_{AB}^b) is only virtual, we cannot simply claim that the region of achievable rates in this virtual channel can be derived directly from the conventional cut-set bound analysis, see [32,39,40], but a careful information-theoretic analysis would be required to identify the exact rate region. However, such analysis is far beyond the scope of this report, and hence, for simplicity reasons, we only conjecture that the eligible source rates ($r_b,\ r_s$) are limited by the conventional CC multiple-access capacity region[15], which can be defined by the following set of mutual information $I(\textbf{.};\textbf{.})$:

$$2r_s + r_b \leq I(x; \mathbf{c}_A^s, \mathbf{c}_B^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b)), \tag{2.12}$$

$$2r_s \leq I(x; \mathbf{c}_A^s, \mathbf{c}_B^s | f(\mathbf{c}_A^b, \mathbf{c}_B^b)), \tag{2.13}$$

$$r_s + r_b \leq I(x; \mathbf{c}_A^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b) | \mathbf{c}_B^s), \tag{2.14}$$

$$r_s + r_b \leq I(x; \mathbf{c}_B^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b) | \mathbf{c}_A^s), \tag{2.15}$$

$$r_s \leq I(x; \mathbf{c}_A^s | \mathbf{c}_B^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b)), \tag{2.16}$$

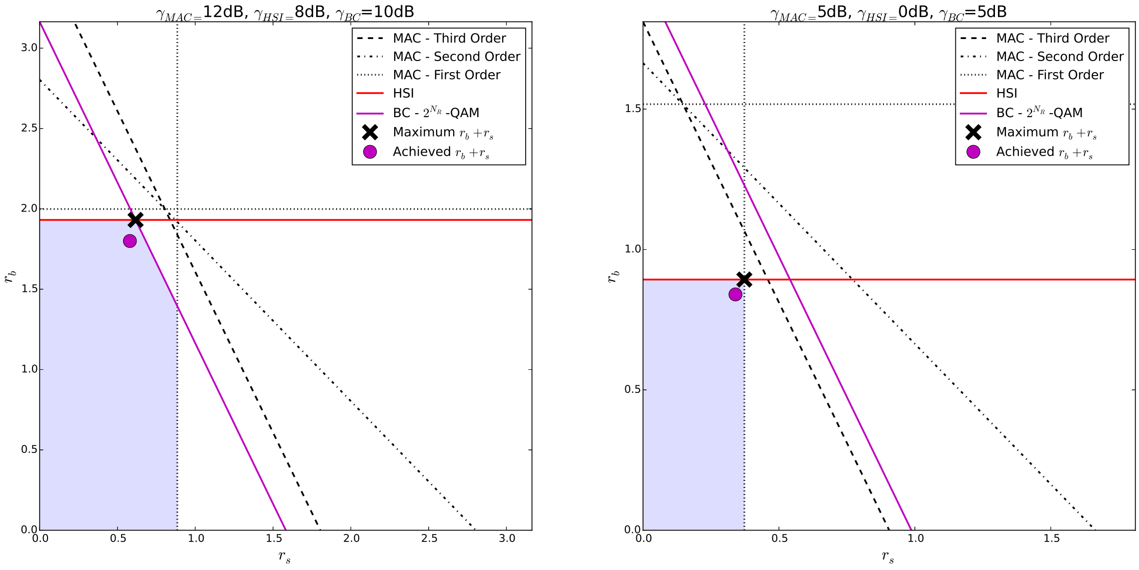
$$r_s \leq I(x; \mathbf{c}_B^s | \mathbf{c}_A^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b)), \tag{2.17}$$

$$r_b \leq I(x; f(\mathbf{c}_A^b, \mathbf{c}_B^b) | \mathbf{c}_A^s, \mathbf{c}_B^s), \tag{2.18}$$

where (2.12), (2.13)-(2.15) and (2.16)-(2.18) are the corresponding third, second and first order cut-set bounds (respectively). As we show later, the CC MAC region (as defined by (2.12)-(2.20)) provides a reasonable estimate of the maximal achievable source transmission rate pair ($r_b,\ r_s$) in WBN.

Now, since the superposed data sub-words $\mathbf{D}_A^s,\ \mathbf{D}_B^s$ are mapped to orthogonal ASK sets (Algorithm 1), the 2nd order cut-set bound $I(x; \mathbf{c}_A^s, \mathbf{c}_B^s | f(\mathbf{c}_A^b, \mathbf{c}_B^b))$ (2.13) is equivalent to $I(x; \mathbf{c}_A^s | \mathbf{c}_B^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b)) + I(x; \mathbf{c}_B^s | \mathbf{c}_A^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b))$ (defined in eqs. (2.16) and (2.17)), which can be proven easily by the chain rule for mutual information of orthogonal signal sets (see [32]). Moreover, due to the assumed WBN system symmetry, the cut-set bounds defined in equations (2.16), (2.17) are identical, and the same is true for the pair of cut-set bounds defined in (2.14), (2.15). Consequently, the CC multiple-access capacity region is completely defined by the four inequalities in (2.12), (2.14), (2.16) and (2.18).

[15] For a detailed discussion on the cut-set bound evaluation of MAC capacity regions see [32,39,40].


γ_{MAC}=12dB, γ_{HSI}=8dB, γ_{BC}=10dB

γ_{MAC}=5dB, γ_{HSI}=0dB, γ_{BC}=5dB

Figure 2.11: Numerically evaluated cut-set bounds (mutual information for finite input constellations) for $(N_b = 2, N_s = 1)$ source constellations and 16-QAM at the relay. The particular SNR conditions are available in the titles of both sub-figures.

Effective HSI channels

In addition to the successful relay decoding, the transmission rate of the basic information stream r_b has to guarantee that the HSI (given by the unintended data sub-words $\mathbf{D}_A^b$ at D_B and $\mathbf{D}_B^b$ at D_A) can be decoded at both destinations (after perfect IC of superposed data codewords $\mathbf{C}_A^s$ at D_B and $\mathbf{C}_B^s$ at D_A from the stored MA phase signal). Note that since we assume that the relay has already successfully decoded $f(\mathbf{D}_A^b, \mathbf{D}_B^b)$ at this step, it can be efficiently exploited in the decoding process, resulting in the following upper-bounds on the source rate r_b:

$$r_b \leq I(z_B^{\mathrm{IC}}; \mathbf{c}_A^b | f(\mathbf{c}_A^b, \mathbf{c}_B^b)), \tag{2.19}$$

$$r_b \leq I(z_A^{\mathrm{IC}}; \mathbf{c}_B^b | f(\mathbf{c}_A^b, \mathbf{c}_B^b)), \tag{2.20}$$

where z_A^{IC}, z_B^{IC} are the efficient HSI observations (after perfect IC) at D_A, D_B (respectively) – see Figure 2.10. Both (2.19), (2.20) are identical in the symmetric WBN system.

Relay broadcast channel

The relay rate r_R has to guarantee that its data word $\mathbf{D}_R = [\hat{\mathbf{D}}_A^s, \hat{\mathbf{D}}_B^s, \hat{f}(\mathbf{D}_A^b, \mathbf{D}_B^b)]$ with length $|\mathbf{D}_R| = 2k_{\mathbf{D}}^s + k_{\mathbf{D}}^b = \mathcal{N}(2r_s + r_b)$ can be sent to both destinations in $\mathcal{N}$ channel uses.

In addition to this, to enable successful decoding of the relay data at both destinations, the relay transmission rate r_R has to be below the CC capacities of the corresponding relay→destination channels. Thus, the relay rate should be set to $r_R = r_b + 2r_s$, which gives us the last two inequalities for $r_b,\ r_s$:

$$r_R = 2r_s + r_b \leq I(y_A; \mathbf{c}_R), \tag{2.21}$$

$$r_R = 2r_s + r_b \leq I(y_B; \mathbf{c}_R), \tag{2.22}$$

where both cut-set bounds (2.21), (2.22) are identical due to the system symmetry.

2.2.4 Transmission rate region for fixed source constellation

To identify the region $(r_b,\ r_s) \in \mathcal{R}^{(N_b,N_s)}$ of eligible transmission rates for a fixed (N_b, N_s) source constellation, we can evaluate numerically[16] the set of relevant cut-set bounds, i.e. (2.12), (2.14), (2.16), (2.18), (2.19) and (2.21), for given SNR conditions in the symmetric WBN system $(\gamma_{\mathrm{MAC}},\ \gamma_{\mathrm{BC}},\ \gamma_{\mathrm{HSI}})$.

An example analysis of the source transmission rate region $\mathcal{R}^{(2,1)}$ for the $(N_b = 2,\ N_s = 1)$ source constellation is visualised in Figure 2.11 for two different SNR setups. The cut-set bound rate region $\mathcal{R}^{(N_b,N_s)}$ is emphasised by the shaded area and the rate pair with maximal $r_b + r_s$ is identified in the figure. We have assumed that the source→relay channels are AWGN (or equivalently that perfect source phase prerotation [11, 41, 42] is implemented) in the numerical evaluation.

2.2.5 Performance of the adaptive encoded system

As shown in Figure 2.11, the performance of the encoded WBN system is, like in the uncoded case, influenced by the immediate SNR conditions (given by $\gamma_{\mathrm{MAC}},\ \gamma_{\mathrm{BC}},\ \gamma_{\mathrm{HSI}}$), and hence an *adaptive modulation-coding scheme* is again of interest. However, before we introduce the *SNR mapping operation* for the encoded system, we show that the cut-set bound rate regions $\mathcal{R}^{(N_b,N_s)}$ (as introduced in the preceding section) provide a reasonable approximation of achievable source transmission rate pairs in the symmetric WBN system. In the whole section we assume (for simplicity reasons) that perfect source phase prerotation [11, 41, 42] is implemented in the system, virtually reverting the source→relay channels to the AWGN case.

[16] The (conditional) probability density function $p(x|\mathbf{c}_A^s, \mathbf{c}_B^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b))$ which is required for the evaluation of particular mutual information in (2.12)-(2.20) can be obtained by a proper marginalisation operation $p(x|\mathbf{c}_A^s, \mathbf{c}_B^s, f(\mathbf{c}_A^b, \mathbf{c}_B^b)) = \Pr\left\{f(\mathbf{c}_A^b, \mathbf{c}_B^b)\right\} \cdot \sum_{\mathbf{c}_A^b, \mathbf{c}_B^b : f(\mathbf{c}_A^b, \mathbf{c}_B^b)} p\left(x|\mathbf{c}_A^s, \mathbf{c}_B^s, \mathbf{c}_A^b, \mathbf{c}_B^b\right)$, where $p(x|\mathbf{c}_A^s, \mathbf{c}_B^s, \mathbf{c}_A^b, \mathbf{c}_B^b)$ is the likelihood function. For more details on the evaluation of CC capacity for finite input constellations see [1, 39, 40].

Numerical simulation of encoded WBN system throughput

We implement all constituent channel encoders in the WBN system (see Figure 2.10) as parallel concatenated turbo codes with random interleavers of length $\mathcal{N} = 10^5$ symbols. By a proper puncturing [43] of the channel encoders' outputs we are able to set the source transmission rates (separately for the basic and superposed streams) to an arbitrary value.

We set the initial source transmission rates to the maximum $r_b + r_s \in \mathcal{R}^{(N_b,N_s)}$ (as evaluated in Figure 2.11) and by proper adjustment of the puncturing operation we successively decrease the source transmission rates $r_b,\, r_s$ until the SER performance drops to zero. As one can see in Figure 2.11, the achieved transmission rate pairs are relatively close to the maximal permissible values, and hence we can consider the cut-set bound rate region $\mathcal{R}^{(N_b,N_s)}$ (as discussed in Section 2.2.4) to be a reasonable approximation of the achievable source transmission rate region.

Maximal throughput of the adaptive modulation-coding scheme

The source nodes $S_A,\, S_B$ in the adaptive modulation-coding WBN system has to be aware of the particular mapping operation:

$$(\hat{\gamma}_{\mathrm{MAC}},\, \hat{\gamma}_{\mathrm{BC}},\, \hat{\gamma}_{\mathrm{HSI}}) \mapsto \left(N_b^{\mathrm{II}},\, N_s^{\mathrm{II}};\, r_b^{\mathrm{II}},\, r_s^{\mathrm{II}}\right), \tag{2.23}$$

providing the set of optimal (in the sense of maximal throughput) constellation/encoder parameters for the given SNR conditions $(\hat{\gamma}_{\mathrm{MAC}},\, \hat{\gamma}_{\mathrm{BC}},\, \hat{\gamma}_{\mathrm{HSI}})$. Note that, in general, all adaptive system parameters are functions of the actual SNR conditions in the system. However, we mostly omit the explicit notation in the following text for a better readability.

The desired adaptive system parameters can be obtained in the following two-step procedure. First of all, we analyse the cut-set bound rate regions $\mathcal{R}^{(N_b,N_s)}$ (as discussed in Section 2.2.4) to identify the maximal encoded system throughput $T_C^{(N_b,N_s)}$ for a fixed source constellation (N_b, N_s):

$$\begin{aligned} T_C^{(N_b,N_s)}(\gamma_{\mathrm{MAC}},\, \gamma_{\mathrm{BC}},\, \gamma_{\mathrm{HSI}}) &= \max_{r_b, r_s \in \mathcal{R}^{(N_b,N_s)}} (r_b + r_s), & (2.24)\\ &= r_b^{\max}(N_b, N_s) + r_s^{\max}(N_b, N_s). & (2.25) \end{aligned}$$

Then, an exhaustive search over the constellation parameters $N_b,\, N_s$ can be performed to find the maximum encoded system throughput $T_C^{\max}$ for the given SNR conditions $(\gamma_{\mathrm{MAC}},\, \gamma_{\mathrm{BC}},\, \gamma_{\mathrm{HSI}})$:

$$T_C^{\max}(\gamma_{\mathrm{MAC}},\, \gamma_{\mathrm{BC}},\, \gamma_{\mathrm{HSI}}) = T_C^{(N_b^{\mathrm{II}},\, N_s^{\mathrm{II}})} = \max_{(N_b,N_s)} T_C^{(N_b,N_s)}, \tag{2.26}$$

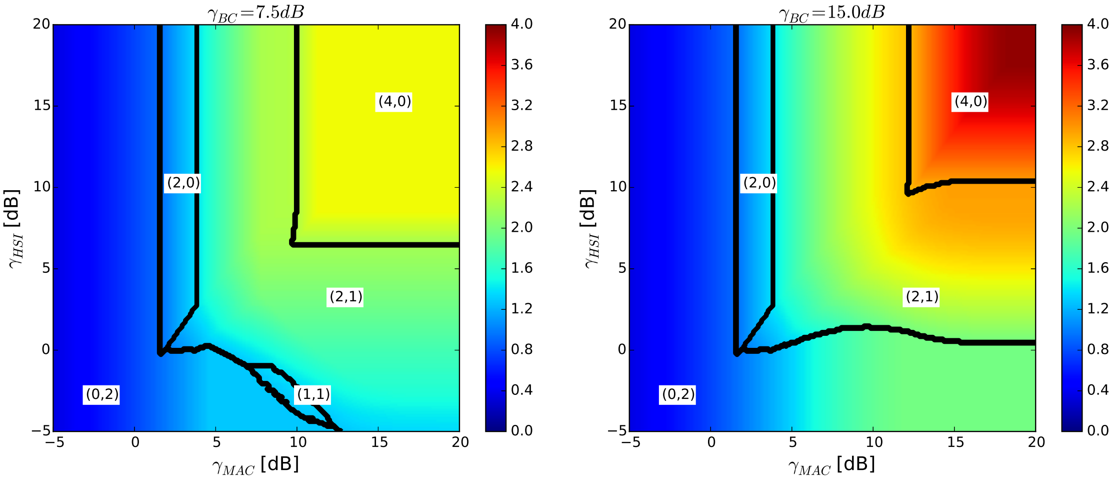


Figure 2.12: Maximal encoded system throughput $T_C^{\max}$ for $\gamma_{\text{BC}} = 7.5\,\text{dB}$ and $\gamma_{\text{BC}} = 15\,\text{dB}$.

where $(N_b^{\text{II}},\, N_s^{\text{II}})$ are the *optimal source constellation parameters*. The corresponding *optimal source transmission rates* (2.25) are given simply by $r_b^{\text{II}} = r_b^{\max}(N_b^{\text{II}},\, N_s^{\text{II}})$, $r_s^{\text{II}} = r_s^{\max}(N_b^{\text{II}},\, N_s^{\text{II}})$.

The SNR mapping regions (including the throughput $T_C^{\max}$) for the adaptive modulation-coding scheme in WBN are evaluated as a function of γ_{MAC}, γ_{HSI} in Figure 2.12 for $\gamma_{\text{BC}} \in \{7.5\,\text{dB},\, 15\,\text{dB}\}$. The optimal source transmission rates r_b^{II}, r_s^{II} of the basic and superposed data streams are depicted in Figures 2.13 and 2.14, respectively. The optimal constellation parameters $(N_b^{\text{II}},\, N_s^{\text{II}})$ are emphasised in all figures.

As one can see in Figure 2.15, the increased reliability of transmissions on all individual channels in WBN (as compared to the uncoded system) translates directly into a significant enhancement of throughput performance in the whole range of observed SNRs. This observation justifies the viability of the proposed channel coding scheme (Figure 2.10).

2.2.6 Conclusions

In this section we have introduced a channel coding scheme for source constellations designed in Section 2.1. Even in the uncoded scenario, the constellations designed in Section 2.1 are able to significantly boost the WBN system throughput, and hence their implementation could be interesting especially in some basic wireless networks, where the channel coding operation is avoided due to simplicity requirements, e.g. sensor networks. Furthermore, in this section we have shown that the proposed source constellation design can be readily extended with appropriately designed channel encoding/decoding operations, significantly improving the reliability (and maximum throughput) of the resulting adaptive modulation-coding scheme.

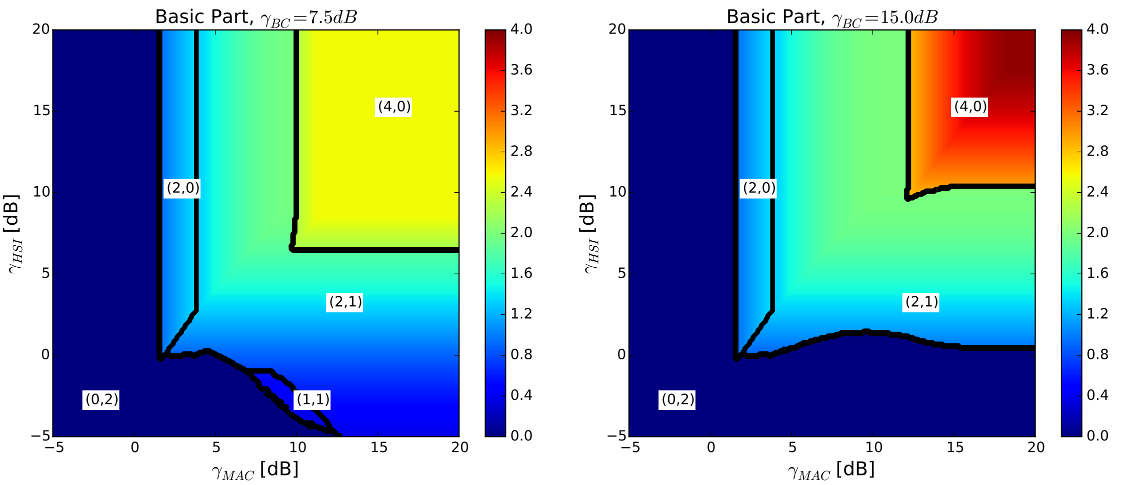


Figure 2.13: Optimal rate r_b^{II} of the basic data stream for $\gamma_{\mathrm{BC}} = 7.5\,\mathrm{dB}$ and $\gamma_{\mathrm{BC}} = 15\,\mathrm{dB}$.

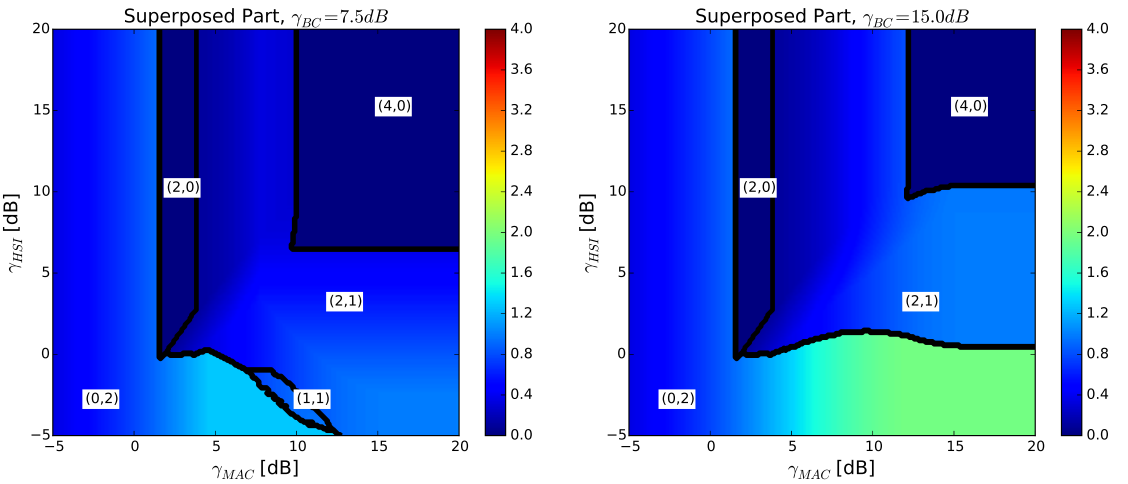


Figure 2.14: Optimal rate r_s^{II} of the superposed data stream for $\gamma_{\mathrm{BC}} = 7.5\,\mathrm{dB}$ and $\gamma_{\mathrm{BC}} = 15\,\mathrm{dB}$.

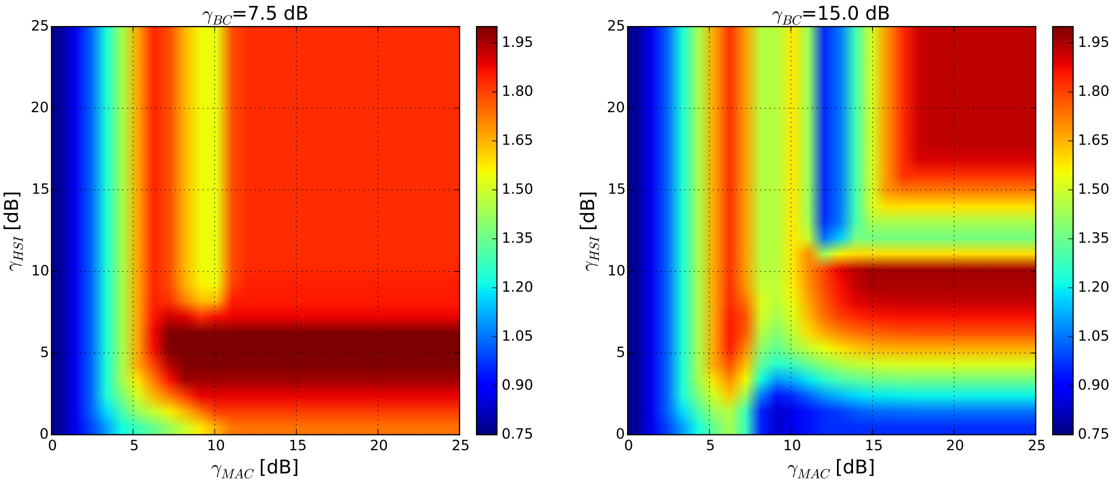


Figure 2.15: Throughput enhancement $\Delta T_C = T_C^{\max} - T_{\mathrm{LB}}$ in the coded WBN system for $\gamma_{\mathrm{BC}} = 7.5\,\mathrm{dB}$ and $\gamma_{\mathrm{BC}} = 15\,\mathrm{dB}$. Note that $0.75 \leq \Delta T_C \leq 2$ in the analysed range of channel SNRs.

In future work, we would like to analyse the robustness of the proposed adaptive system in a full HW setup, where the impact of several parameters (e.g. MAC channel fading, asymmetry of channel gains, imperfect source phase pre-rotation or a direct source$\rightarrow$destination visibility) can significantly affect the system performance and possibly even require modification of the adaptive system design.

3 Compute and forward based NCM

3.1 Non-cooperative CaF

Network coding has been a very promising topic since introduced by Ahlswede et al. in [44]. The concept of network coding in wired networks is very well investigated and first implementations for industrial usage are emerging [45, 46]. Wireless networks on the other hand are still subject of intensive research. The properties of the wireless channel allow network coding on different layers. The superposition property can be exploited by the relays and enables WPLNC. Since the introduction of physical layer network coding [47], different schemes have been developed, e.g. [41, 48–50]. A framework that has drawn a lot of attention is compute-and-forward [22]. It uses lattice codes and exploits their structure to allow the decoding of linear equations of codewords without decoding the codewords themselves. This enables the relaying nodes to decode the equations at higher rates than it would be possible by decoding the individual messages, e.g. by decode-and-forward. A further important advantage compared to e.g. amplify-and-forward is the possibility to avoid noise accumulation, which reduces the performance in large networks. Also this has a lot of advantages and high rates are achievable, this comes with the need for network diversity. The final destination needs to collect enough independent equations to jointly decode the transmitted data.

The performance of compute-and-forward highly depends on the alignment of the channel coefficients with the coefficients of the decoded equation. Several algorithms have been proposed to solve for the optimal equation in terms of achievable computation rate [51–53]. While most of these algorithms perform a local optimisation, they ignore the network structure and the possible linear dependence of the equations at the final destination. This will result in outages because the destination is not able to decode the messages if it has not enough linear independent equations. This is a very serious problem in real applications, especially in large networks with several hops because retransmissions from all sources to the destination are not applicable. The delay might be extremely large and the signalling will pollute the network. This problem can be overcome by allowing cooperation between nodes [53]. In realistic setups however, this might not be applicable due to the large signalling overhead to allow this kind of cooperation. Further, it is not very robust against channel state changes. Therefore, the question arises if it is possible to exploit the network structure such that no cooperation between nodes is needed except for a network initialisation phase.

In this section we introduce some non-cooperative schemes which enforce linear independence of the decoded equations at the destination. We compare the performance of these schemes in multi-source multi-relay networks and show that correlation between channel coefficients plays a crucial role. We find that correlation can decrease the achievable sum-rate of compute-and-forward by 1.5 bit/cu whereas our proposed schemes are robust against correlation and can achieve twice the sum-rate. The section is organised as follows. We define the system model in Section 3.1.1 and introduce the different relaying schemes in Section 3.1.4. We discuss the performance of these schemes with the help of simulation results in Section 3.1.5 and conclude this work in Section 3.1.7.

Notation

Let $\log^+(x) \triangleq \max\{0, \log(x)\}$ and $\mathbb{F}_p$ a finite field of size p, where p is a prime. We denote by x^T the transpose of vector x and by e_i the unit vector with a one at position i and zeros elsewhere.

In the following we recall some lattice definitions that are used throughout the paper. For further details on lattice codes see [17, 48, 54, 55]. An n-dimensional lattice $\Lambda \subset \mathbb{R}^n$ is a group under addition with generator matrix $G \in \mathbb{R}^{n\times n}$, i.e. $\Lambda = \{Gc : c \in \mathbb{Z}^n\}$. A lattice quantiser is a mapping $Q_\Lambda : \mathbb{R}^n \to \Lambda$ that maps a point x to the nearest lattice point in Euclidean distance, i.e.,

$$Q_\Lambda(x) = \arg\min_{\lambda\in\Lambda} \|x - \lambda\|. \tag{3.1}$$

Let the modulo operation with respect to the lattice Λ be defined as $[x] \bmod \Lambda = x - Q_\Lambda(x)$. We call $\mathcal{V} = \{x : Q_\Lambda(x) = 0\}$ the fundamental Voronoi region of the lattice Λ and denote by Vol$\mathcal{V}$ the volume of $\mathcal{V}$.

Two lattices Λ_C and Λ_F are called nested, if $\Lambda_C \subseteq \Lambda_F$. We call Λ_C the coarse lattice and Λ_F the fine lattice. A nested lattice code $\mathcal{L}$ is formed by taking all of the points of the fine lattice Λ_F in the fundamental Voronoi region $\mathcal{V}_C$ of the coarse lattice Λ_C, i.e. $\mathcal{L} = \Lambda_F \cap \mathcal{V}_C$. The rate of a nested lattice code is given by

$$r = \frac{1}{n}\log|\mathcal{L}| = \frac{1}{n}\log\frac{\mathrm{Vol}\mathcal{V}_C}{\mathrm{Vol}\mathcal{V}_F}. \tag{3.2}$$

3.1.1 System model

We investigate an $L \times M$ relay network consisting of L source nodes, M relay nodes and one destination node as depicted in Figure 3.1 and in more detail in Figure 3.2. Each source node ℓ has a message w_ℓ that has to be transmitted to the destination node. Because there are no direct links between the source nodes and the destination, the transmission has to use the help of several relays. For the ease of simplicity we assume that all M relays are used. The relays can apply several relaying strategies which are explained in detail in Section 3.1.4. The used relaying strategy is common to all relays and a system

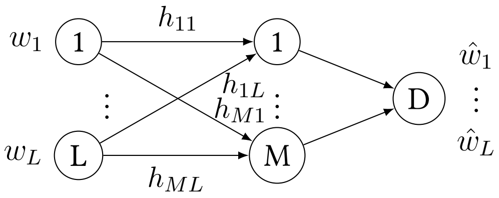


Figure 3.1: System model of a $L \times M$ relay network

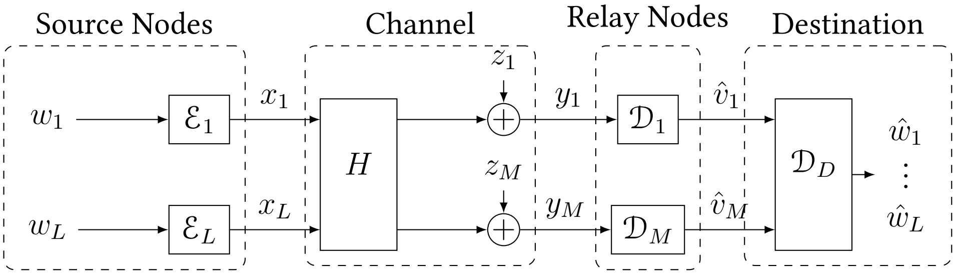


Figure 3.2: System model of a $L \times M$ relay network

design parameter. We will investigate which relaying strategy is best to use for certain system parameters.

We want to stress that we assume no cooperation between the relays because this would drastically increase the complexity in realistic scenarios. The cost for network providers in terms of infrastructure complexity is much less when the nodes are independent. For example we have less signalling overhead. Further, node independence increases the robustness of the network against topology changes.

We will focus our work on the communication between source and relay nodes and the decoding at the relay nodes. There might be other layers or networks after the relays which are beyond the scope of the investigation. Therefore, we assume that the channels between the relays and the destination are bit-pipes with large enough capacity. They are error-free and do not interfere with each other. This might be achieved by FDMA or similar techniques.

3.1.2 General definitions

Definition 3.1 (Messages)**.** Each source node ℓ chooses a length-k message vector $w_\ell \in \mathbb{F}_p^k$ i.i.d. from a uniform distribution over the index set $\{1, 2, \ldots, 2^{\lfloor nR_\ell \rfloor}\}$. Because we will build linear combinations of these messages, we zero-pad them to a common length $k \triangleq \max_\ell k_\ell$.

Definition 3.2 (Encoders)**.** Each source node is equipped with an encoder,

$$\mathcal{E}_\ell : \mathbb{F}_p^k \to \Lambda_F \cap \mathcal{V}_C, \tag{3.3}$$

that maps length-k messages over the finite field to length-n real-valued codewords of a lattice code $\mathcal{L} = \Lambda_F \cap \mathcal{V}_C$, i.e. $x_\ell = \mathcal{E}_\ell(w_\ell)$. Each codeword is subject to the power constraint $\|x_\ell\|^2 \leq nP$. Therefore, Λ_C is chosen such that the second moment of Λ_C equals P.

Definition 3.3 (Channel model)**.** Each relay receives a noisy linear combination of the transmitted signals through the channel, i.e.,

$$y_m = \sum_{\ell=1}^{L} h_{m\ell} x_\ell + z_m, \tag{3.4}$$

where $h_{m\ell} \in \mathbb{R}$ are the channel coefficients and z_m is i.i.d. Gaussian noise, i.e. $z_m \sim \mathcal{N}(0, I_n)$. Let $h_m = (h_{m1}, \ldots, h_{mL})^T$ denote the vector of channel coefficients to relay m and let $H = \{h_{m\ell}\}$ denote the entire channel matrix. With this notation the m-th row in H is h_m^T.

Definition 3.4 (Desired equations)**.** The goal of each relay is to reliably recover a linear combination of lattice codewords

$$v_m = \left[\sum_{\ell=0}^{L} a_{m\ell} x_\ell\right] \bmod \Lambda_C. \tag{3.5}$$

The desired equation is represented by a coefficient vector $a_m = (a_{m1}, \ldots, a_{mL})^T$.

Definition 3.5 (Message rate)**.** The message rate of each source node is the length of its message normalised by the number of channel uses,

$$R_\ell = \frac{k_\ell}{n} \log_2 p. \tag{3.6}$$

Definition 3.6 (Computation rate)**.** The computation rate $\mathcal{R}(h_m, a_m)$ is achievable if for any $\varepsilon > 0$ and n large enough, there exist encoders and decoders, such that all relays can recover their desired equations with average probability of error ε so long as the underlying message rates R_ℓ satisfy

$$\forall \ell \in \{1, \ldots, L\} : R_\ell < \min_{m: a_{m\ell} \neq 0} \mathcal{R}(h_m, a_m). \tag{3.7}$$

3.1.3 Decoding at the destination

The destination node receives M linear combinations of the lattice codewords from the relay nodes, i.e.,

$$y_D = A \cdot x, \tag{3.8}$$

where $x = (x_1, \dots, x_L)^T$ is the vector of transmitted lattice codewords from all source nodes and

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1L} \\ a_{21} & a_{22} & \cdots & a_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{ML} \end{pmatrix} \tag{3.9}$$

is the coefficient matrix of the linear combinations. Please note that the m-th row in A is equal to the coefficient vector a_m of the linear combination decoded at relay m. The destination can solve the system of linear equations in Equation (3.8) via matrix inversion if the coefficient matrix A has full rank, i.e. rank$(A) \geq m$. Otherwise an outage occurs.

Definition 3.7 (Outage probability)**.** The outage probability P_{out} is the probability that the destination node is not able to decode each single codeword x_ℓ transmitted by source node ℓ, $\ell \in \{1, \dots, L\}$.

Definition 3.8 (Achievable sum-rate)**.** The achievable sum-rate is defined as the goodput of all source nodes to the destination,

$$R_{\text{sum}} = (1 - P_{\text{out}}) \cdot \sum_{\ell=1}^{L} R_\ell, \tag{3.10}$$

where P_{out} is the outage probability.

3.1.4 Relaying strategies

In this subsection we introduce the relaying strategies. All of them are based on compute-and-forward as introduced in [22]. This means each relay decodes a linear combination of lattice codewords with coefficient vector a_m. The achievable computation rate at relay m is given by

$$\mathcal{R}(h_m, a_m) = \frac{1}{2} \log_2^+ \left(\frac{1}{a_m^T \cdot G_{\text{CF}} \cdot a_m} \right), \tag{3.11}$$

where

$$G_{\text{CF}} = I_{L \times L} - \frac{P}{1 + P\|h_m\|^2} \cdot h_m h_m^T. \tag{3.12}$$

Compute-and-Forward is able to achieve a high computation rate due to the fact that it decodes a linear combination of messages and not each single message like for example decode-and-forward. However, this does not take the network structure into account. In a large network with multiple hops we get outages due to linear dependent equations at the destination. Even for a single hop with multiple relays the outage probability is very high (see Figure 3.3). This motivates us to modify the optimisation problem of finding the best equation such that the network structure is exploited. Therefore, we introduce some modified compute-and-forward schemes which differ in the way a_m is chosen.

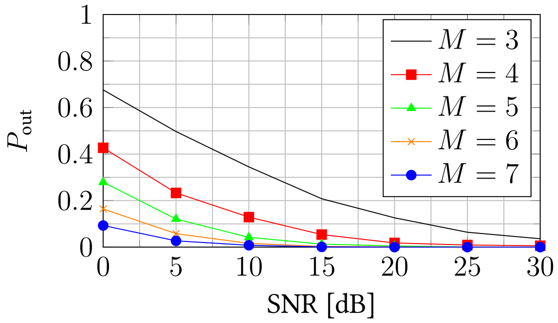


Figure 3.3: Probability of rank failure in classical compute-and-forward for 3 source nodes and M relay nodes.

Classical compute-and-forward

With the term classical compute-and-forward we refer to the unaltered scheme introduced in [22]. In the classical compute-and-forward scheme each relay solves the following optimisation problem individually to maximise the achievable computation rate,

$$R_{\text{CF},m}^{\text{CCF}} = \max_{a_m \in \mathbb{Z}^L \setminus \{0\}} \mathcal{R}(h_m, a_m). \tag{3.13}$$

This can be solved by several algorithms, e.g. [51–53].

Because we assume no cooperation between the relays, it is not guaranteed that the relay nodes decode linear independent linear combinations. Therefore, the destination node cannot always decode all codewords. The outage probability P_{out} in this case is the probability that the coefficient matrix A does not have full rank, i.e.,

$$P_{\text{out}} \triangleq \Pr\{\text{rank}(A) < m\}. \tag{3.14}$$

A possible method to reduce the outage probability is to decode only non-zero equations a_m [22, Sec. X]. However, this will significantly reduce the achievable rate and does not guarantee zero outage probability.

In real world applications we have to ensure that the transmitted data is decodable at the destination. In a point-to-point channel with outages this is usually done by an ARQ protocol. In the compute-and-forward framework one would have to re-transmit the codewords from all sources. In multi-hop networks this is usually not applicable due to large delays and large overhead.

Single user decoding

By single user decoding we refer to a special case of compute-and-forward, where relay m decodes the linear combination $a_m = e_m$. This means each relay decodes only a single codeword and treats all other signals as noise. Because each relay decodes a different codeword, it is guaranteed that the coefficient matrix A at the destination node has full rank. Therefore, we do not have outages, i.e. $P_{\text{out}} = 0$. However, this comes at the cost of achievable computation rate because we choose a sub-optimal coefficient vector in Equation (3.13). The achievable computation rate at relay m is given by

$$R_{\text{CF},m}^{\text{SU}} = \mathcal{R}(h_m, e_m). \tag{3.15}$$

Subspace compute-and-forward

In the following we introduce subspace compute-and-forward. We reduce the feasible set of possible coefficient vectors in the optimisation problem in Equation (3.13) for each relay such that the subsets are linearly independent. We get the following optimisation problem at relay m

$$R_{\text{CF},m}^{\text{SCF}} = \max_{a_m \in \mathcal{S}_m} \mathcal{R}(h_m, a_m), \tag{3.16}$$

where $\mathcal{S}_m \subset \mathbb{Z}^L$. For the construction of the subsets we restrict ourselves to the case where $L = M$, i.e. the number of relay nodes equals the number of source nodes. It is obvious that in the case of $M > L$ we get $M - L$ linear dependent equations at the destination which are not necessary for decoding. This becomes a problem of relay selection and is beyond the scope of this paper.

Let $B \in \mathbb{Z}^{L \times L}$ be a basis matrix for $\mathbb{Z}^L$. We define B_m to be a sliced version of B containing only the first m rows of B. The subset $\mathcal{S}_m$ is then constructed by

$$\mathcal{S}_m = \{B_m^T \cdot \beta_m : \beta_m \in \mathbb{Z}^m, \beta_{mm} \neq 0\}. \tag{3.17}$$

The achievable computation rate at relay m is the solution to the following optimisation problem

$$R_{\text{CF},m}^{\text{SCF}} = \max_{\substack{\beta_m \in \mathbb{Z}^m \\ \beta_{mm} \neq 0}} \mathcal{R}(h_m, B_m^T \beta_m), \tag{3.18}$$

which can be written as

$$R_{\text{CF},m}^{\text{SCF}} = \max_{\substack{\beta_m \in \mathbb{Z}^m \\ \beta_{mm} \neq 0}} \frac{1}{2} \log_2^+ \left(\frac{1}{\beta_m^T \cdot G_{\text{SCF}} \cdot \beta_m} \right), \tag{3.19}$$

where

$$G_{\text{SCF}} = B_m B_m^T - \frac{P}{1 + P\|h_m\|^2} \cdot B_m h_m h_m^T B_m^T. \tag{3.20}$$

Please note that the optimisation problem has the same structure as the one for classical compute-and-forward (see Equation (3.13)). If a sorted channel vector h_m is assumed, then the constraint sets are also equivalent (see [51] for details). Therefore, we can use the same algorithms to solve this optimisation problem.

Hierarchical compute-and-forward

Hierarchical compute-and-forward has been introduced independently and from different points of view in [56] and [57]. This scheme fixes the equations that are decoded at each relay independently from the channel realisations. Therefore, the equation coefficients can be chosen such that linear independence at the destination is guaranteed. This comes at the price of a lower performance because the channel coefficients and the equation coefficients are not well aligned. The key technique used to improve the performance is interference cancellation. The decoder at relay m decodes an auxiliary equation $\tilde{a}_m$ and subtract this from the received signal to create a new virtual channel

$$\tilde{h}_m = h_m - \eta \tilde{a}_m. \tag{3.21}$$

The auxiliary equation has to be chosen such that the decoding performance of the desired equation a_m is increased for this virtual channel. The coefficient $\eta \in \mathbb{R}$ has to be chosen such that the mean squared error between the virtual channel $\tilde{h}_m$ and the desired coefficients a_m is minimised. To obtain the optimal auxiliary equation one has to solve the following optimisation problem at relay m,

$$R_{\text{CF},m}^{\text{HCF}} = \max_{\tilde{a} \in \mathcal{A}_m} \min\{\mathcal{R}(h_m, \tilde{a}_m), \mathcal{R}(\tilde{h}_m, a_m)\}, \tag{3.22}$$

where $\mathcal{A}_m = \{\tilde{a}_m : \mathcal{R}(h_m, \tilde{a}_m) > \mathcal{R}(h_m, a_m)\}$ is the set of all auxiliary equations which improve the performance. So far the optimisation problem in Equation (3.22) lacks an efficient algorithm to solve this problem. It is basically a combinatorial problem which can be solved by an exhaustive search with high complexity.

Please note that hierarchical compute-and-forward can be combined with single user decoding as well as subspace compute-and-forward to improve the performance of the respective scheme. However, one has to keep in mind that this will increase the computational complexity for solving the respective optimisation problems drastically and might not be worth the performance increase.

3.1.5 Simulation results

In this section we provide simulation results showing the performance of the relaying schemes introduced above. We use the algorithm in [51] to obtain the coefficients for classical and subspace compute-and-forward. Further we use an exhaustive search for hierarchical compute-and-forward. Please note that we measure the performance solely in terms of achievable sum-rate. Some practical concerns about signalling overhead and delay are already raised in Section 3.1.4. In addition to the non-cooperative schemes, we also plot the achievable sum-rate for compute-and-forward with cooperation between the relays as an upper bound. Therefore, we use the algorithm provided in [53].

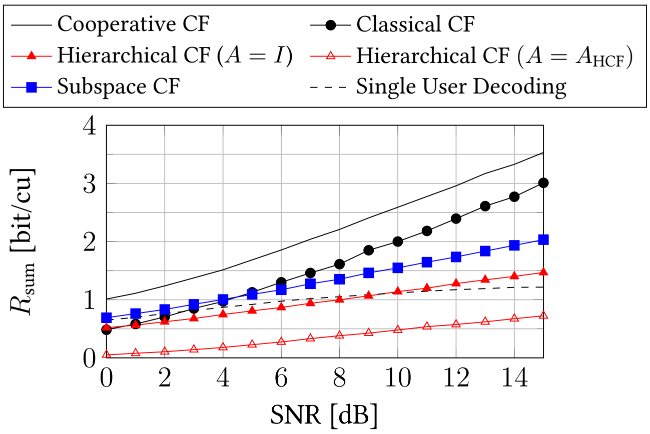


Figure 3.4: Achievable sum-rate for i.i.d. Gaussian channel coefficients. 10,000 channel matrices per SNR value.

3.1.6 No correlation

In the following we compare the achievable sum-rate of the different schemes in a 3×3 relay network, where the channel coefficients are i.i.d. according to a Gaussian distribution, i.e. $h_{m\ell} \sim \mathcal{N}(0, 1)$. As one can see in Figure 3.4 the classical non-cooperative compute-and-forward scheme achieves approximately half a bit less sum-rate than the cooperative scheme. The subspace compute-and-forward strategy performs well for low SNR but losses its performance benefit in the high SNR regime.

The performance of hierarchical compute-and-forward highly depends on the chosen desired equations as one can see in Figure 3.4 where we plotted the achievable sum-rate for two different desired coefficient matrices, i.e. $A = I_{3\times3}$ and

$$A = A_{\mathrm{HCF}} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}. \tag{3.23}$$

A good choice for the desired coefficient matrix is $A = I$. One might think that this should results at least in the same achievable sum-rate as single user decoding. Unfortunately this is not true as one can see from Equation (3.7). The achievable computation rate of relay m only constraints the rate of the source nodes whose codewords are decoded with a non-zero coefficient. If relay m decodes with $a_m = e_i$ it would only effect the achievable rate of source node i. If relay m decodes additionally an auxiliary equation with more non-zero entries, the computation rate of that auxiliary equation effects more relays. Because each source node is constrained by the minimum achievable computation rate of all relays, it is possible that the sum-rate is reduced by using an auxiliary equation at a relay although the individual computation rate at the relay benefits from using an auxiliary equation. This is one side-effect of having no cooperation between the relays.

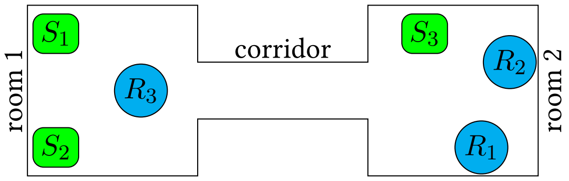


Figure 3.5: 3×3 indoor relay network.

The optimal desired coefficient matrix A for hierarchical compute-and-forward for a given channel distribution is still an open question and will be subject of future research.

Correlation

In this subsection we want to show the influence of correlation on the achievable sum-rate. Therefore, we model the channel between source nodes and relay nodes by the Kronecker model,

$$H = R_R^{1/2} \cdot W \cdot R_T^{1/2}, \tag{3.24}$$

where $W \sim \mathcal{N}(0, I_{M\times L})$ and R_R and R_T are the receive and transmit correlation matrices, respectively. For the simulations we assume a 3×3 relay network with different correlation scenarios:

- strong correlation between two sources and two relays:

$$R_R = \begin{pmatrix} 1.0 & 0.9 & 0.1 \\ 0.9 & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \quad R_T = \begin{pmatrix} 1.0 & 0.9 & 0.1 \\ 0.9 & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \tag{3.25}$$

- weak correlation between two sources and strong correlation between two relays:

$$R_R = \begin{pmatrix} 1.0 & 0.9 & 0.1 \\ 0.9 & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \quad R_T = \begin{pmatrix} 1.0 & 0.1 & 0.1 \\ 0.1 & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \tag{3.26}$$

Equation (3.25) is motivated by an indoor set-up as depicted in Figure 3.5.

As one can see in Figure 3.6 the correlation in the network plays an important role. Strong correlation even between only two source nodes or relay nodes significantly decreases the achievable sum-rate of the classical compute-and-forward scheme (compare to Figure 3.4 without correlation). This is due to the fact that a strong correlation increases the probability that the relay nodes will decode a linear dependent equation. Subspace compute-and-forward and single user decoding however are robust against correlation and do not show a decrease in performance. In fact subspace compute-and-forward shows a superior performance for high correlation.

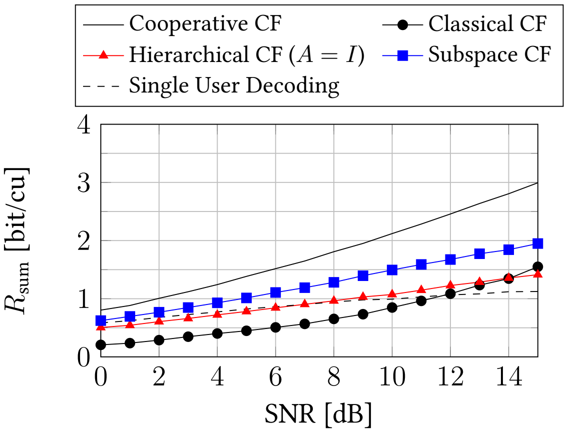


Figure 3.6: Strong receive and transmit correlation between source 1 and 2 as well as relay 1 and 2 according to Equation (3.25).

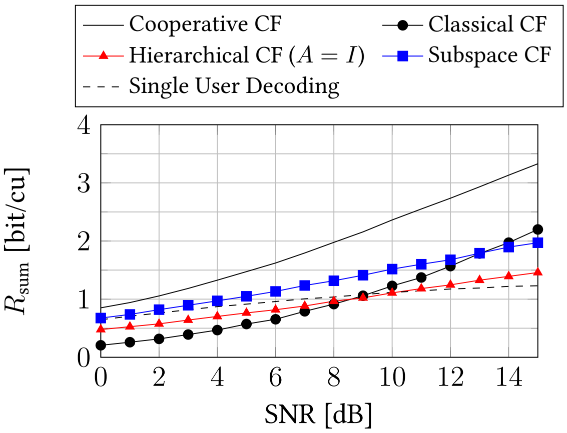


Figure 3.7: Strong receive and weak transmit correlation according to Equation (3.26).

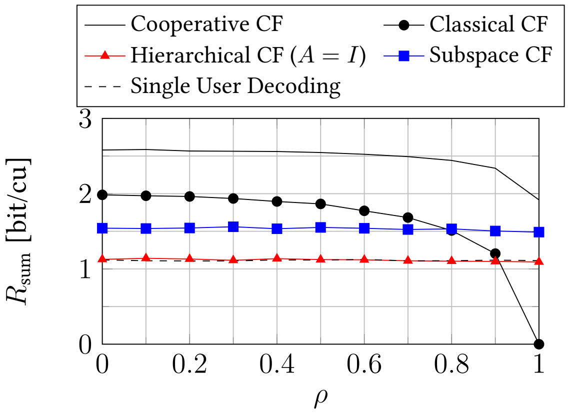


Figure 3.8: Achievable sum-rate with receive correlation ρ according to Equation (3.27).

To get a better impression on the dependency between achievable sum-rate and correlation, we plot the achievable sum-rate over the correlation in Figure 3.8. We consider two scenarios:

- receive correlation between relay 1 and relay 2, i.e.,

$$R_R = \begin{pmatrix} 1.0 & \rho & 0.1 \\ \rho & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \; R_T = \begin{pmatrix} 1.0 & 0.1 & 0.1 \\ 0.1 & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \tag{3.27}$$

- transmit correlation between source 1 and source 2, i.e.,

$$R_R = \begin{pmatrix} 1.0 & 0.1 & 0.1 \\ 0.1 & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \; R_T = \begin{pmatrix} 1.0 & \rho & 0.1 \\ \rho & 1.0 & 0.1 \\ 0.1 & 0.1 & 1.0 \end{pmatrix} \tag{3.28}$$

One can see in Figure 3.8 that a large correlation factor decreases the performance of classical compute-and-forward drastically. On the other hand it can be seen that the non-cooperative schemes show a stable performance.

3.1.7 Conclusion

Because cooperation is an obstacle in large networks we focused on non-cooperative versions of compute-and-forward and proposed new strategies for choosing the desired equations at the relay nodes. We show that the choice of these equations is essential for the performance of the complete network and analysed the impact of correlated signals in the network on the achievable sum-rate. The results can be summarised as follows:

- In networks with several hops, where re-transmissions from source to destination are not applicable, one should use subspace compute-and-forward. The decrease of achievable sum-rate compared to classical compute-and-forward is approximately 1 bit/cu for an SNR of 10 dB.

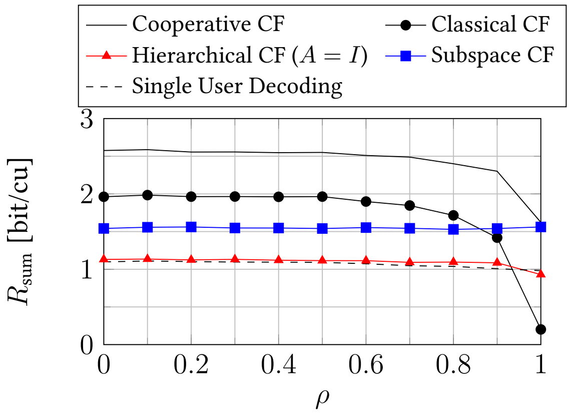


Figure 3.9: Achievable sum-rate with transmit correlation ρ according to Equation (3.28).

- Hierarchical compute-and-forward can increase the performance of subspace compute-and-forward as well as single user decoding in the high SNR regime but comes at the cost of high computational complexity.
- Correlation plays an important role on the performance of classical compute-and-forward and can reduce the achievable sum-rate to 0 bit/cu. Subspace compute-and-forward, hierarchical compute-and-forward as well as single user decoding are robust against correlation.
- If a high correlation between two or more nodes in the network occurs, subspace compute-and-forward shows a superior performance.

As one can see subspace compute-and-forward looks like the winning strategy. It has no increase in computational complexity compared to classical compute-and-forward. It is robust against correlation and has a good performance in almost all scenarios.

3.2 A multi-level framework for lattice network coding

Nazer and Gastpar's work on compute-and-forward (CaF) generalises the traditional WPLNC of a two-way-relay channel to wireless multiuser relay networks by utilising structured nested lattice codes, giving essentially an information-theoretic approach for improving the network throughput and for the network energy saving compared to the traditional routing. Especially to dense wireless cloud networks, the theory behind CaF is appealing, and it is of great interest to investigate the practical design approach based on CaF. We formulated a general algebraic framework, termed multilevel lattice network coding (MLNC), which subsumes the CaF and lattice network coding (LNC), and inherently provides the engineering-applicable design guideline for wireless cooperative networks. MLNC lays the theoretical foundations for solving the ring-based LNC problem in practice, with greatly reduced decoding complexity. It generalises the traditional LNC and CaF by exploiting the rich ring features over the algebraic module, providing a practically feasible solution to the open problem of LNC and CaF. The technique proposed breaks the bottleneck of high-throughput transmission with feasible complexity restrictions over CaF and LNC, which is the most powerful way so far available to manage the complexity, and hence is promising to be engineering-applicable in the dense wireless cloud network.

3.2.1 Introduction

There has recently been a resurgence in research on lattice codes for wireless communications, as a result of two recent developments. The first is recent work [16,58] which has shown that lattice codes with lattice decoding are capable of approaching channel capacity. The second is their application to WPLNC [47] for ultra-dense wireless multihop networks [59]. In particular Nazer and Gastpar have developed CaF [22], which applies structured nested lattice codes to WPLNC for multiuser relay networks. However it is difficult to increase the transmission rate using previous lattice constructions such as construction A lattices, since this requires linear channel codes over large finite fields, for which the decoding complexity is typically unaffordable. In this section, we lay the foundations for a multilevel structure for lattice codes, and uses it to introduce a general lattice construction approach and two multistage decoding approaches which greatly simplify decoding, and which can exploit iterative techniques to approach capacity.

Previous work, e.g. in [49, 60, 61], has given LNC design guidelines when quotient lattices are constructed from existing channel codes using complex construction A. In this section, we consider a multilevel structure for lattice network coding, which provides a practical solution to the ring-based network coding problem. We also propose an efficient lattice construction approach (which we term the elementary divisor construction (EDC)) based on the theorems developed, which also subsumes the most important previous lattice constructions. The EDC lattice has a multilevel algebraic structure, and is well suited for multistage decoding. Note that the recently proposed product construction [62] used in CaF is a special case of EDC. The EDC approach is a straightforward result of the theoretic framework developed in Section 3.2.3. We give explicit representation of the generator matrix for the EDC lattice, propose a new concept of the primary sublattice, and derive the nominal coding gain and kissing numbers for the EDC lattice in all forms. The main contributions of this section are summarised below:

1. We develop a generic multilevel lattice network coding scheme based on some algebraic theorems. This approach keeps beneficial compatibility of the traditional LNC scheme, whereas enabling more flexible coding design techniques. Note that MLNC makes also no particular assumption about the structure of the underlying nested lattice code.

2. We propose a novel lattice network decoding approach based on MLNC, termed layered integer forcing (LIF), which

  - improves the overall throughput for network coding with greatly reduced decoding complexity.
  - decodes lattices which are no longer a vector space.
  - allows flexible linear labelling design for additional performance enhancement.

3. We develop a modified Viterbi algorithm which implements LIF.
4. Building on the algebraic framework developed for MLNC, we present a novel lattice construction approach (EDC approach), show its good structure properties, e.g. the explicit form of the generator matrix, in reducing the decoding complexity, and derive its nominal coding gain and kissing numbers. Mathematically we also prove that EDC lattices subsume the most important complex lattice constructions.
5. We propose a soft detector specifically designed for EDC lattices (as an alternative to LIF for decoding EDC lattices). We evaluate its non-binary extrinsic information transfer characteristics, and propose an iterative multistage decoding approach for EDC lattices, which shows a substantial improvement in decoding performance.
6. We show how multistage detection, iteration-aided multistage detection, and LIF can be applied to MLNC. We also show, by simulation, that iterative decoding performs better than the Viterbi detection approach used in the traditional LNC. This provides the basis for further work, and opens a new research area of iterative decoding for lattice network coding.

The remainder of this paper is organised as follows. In Section 3.2.2 we review some algebraic preliminaries which will be useful in setting up our multilevel framework. Section 3.2.3 studies the algebraic properties of MLNC and presents the practically feasible encoding and decoding solutions. Section 3.2.4 presents a new general lattice construction approach based on MLNC theorems developed and proves that it subsumes some important lattice constructions that have been widely known. Section 3.2.5 presents the soft detector for MLNC and studies the iterative decoding and multistage decoding approaches designed for MLNC. Section 3.2.6 presents the simulation results based on different decoding modes. Section 3.2.7 concludes the paper and presents the future work.

Notations

Notations used throughout this paper are defined as follows. $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{C}$ denote the fields of natural numbers, integers and complex numbers, respectively. $\mathbb{F}_q$, $q > 1$, $q \in \mathbb{Z}$ denotes the finite field of size q. $\mathbb{F}_{\mathbf{q}}^n$ denotes an n-tuple finite field where the field size for the i^{th} dimension $i \in \{1, 2, \cdots, n\}$ is determined by $q_i \in \mathbb{Z}$. We also use boldface lower-case to denote a vector, i.e. $\mathbf{a} = [a^1, a^2, \cdots, a^n]$. $\mathbf{V}^{\backslash i} \triangleq [V^1, V^2, \cdots, V^{i-1}, V^{i+1}, \cdots, V^n]$ represents a set including all elements except the i^{th} one. The upper-case letter, e.g. V, represents a random variable and its realisation is denoted by the lower-case v. The direct sum and direct product are denoted as $\oplus$ and $\times$, respectively.

3.2.2 Algebra preliminaries

We present some definitions and theorems in abstract algebra, which can be found in relevant textbooks, e.g. [63].

Ideal and principal ideal domain

Let R be a commutative ring with identity 1, and $R^* = R\backslash 0$. A *unit* $\mathcal{U}(R)$ in R refers to any element x in R such that $xr = rx = 1$ for some $r \in R$. Any root of unity in a ring R is a unit. An element x in R is called a *zero divisor* $\mathcal{Z}(R)$ if $xr = rx = 0$ for some $r \in R^*$. An element $p \in R$, $p \notin \mathcal{Z}(R)$, $p \notin \mathcal{U}(R)$, is called a *prime* in R when $p \mid ab$ for some $a, b \in R^*$, then either $p \mid a$ or $p \mid b$.

An *ideal* $\mathcal{I}$ of R is a non-empty subset of R that is closed under subtraction (which implies that $\mathcal{I}$ is a group under addition), and is defined by:

1. $\forall a, b \in \mathcal{I}$, $a - b \in \mathcal{I}$.

2. $\forall a \in \mathcal{I}$, $\forall r \in R$, then $ar \in R$ and $ra \in R$.

If $A = \{a_1, \cdots, a_m\}$ is a finite non-empty subset of R, we use $\langle a_1, \cdots, a_m\rangle$ to represent the ideal generated by A, i.e.

$$\langle a_1, \cdots, a_m\rangle = \{a_1 r_1 + \cdots + a_m r_m : r_1, \cdots, r_m \in R\}$$

Note that R has at least two ideals $\{0\}$ and $\{R\}$.

An ideal $\mathcal{I}$ of R is said to be *proper* if and only if $1 \notin \mathcal{I}$. An ideal $\mathcal{I}_{\max}$ is said to be *maximal* if $\mathcal{I}_{\max}$ is a proper ideal and the only ideals that include $\mathcal{I}_{\max}$ are R and $\mathcal{I}_{\max}$ itself. We say that an equivalence relation $a \sim b$ on the set R is defined by $\mathcal{I}$ if and only if $a - b \in \mathcal{I}$.

An ideal $\mathcal{I}$ of R is *principal* if $\mathcal{I}$ is generated by a single element $a \in \mathcal{I}$, written as $\mathcal{I} = \langle a\rangle$. A *principal ideal ring* is a ring whose every ideal is principal. If R is a principal ideal ring without zero divisors, then R forms an ideal domain, and more precisely, a *principal ideal domain* (PID). Examples of PIDs include the ring of integers, the ring of Gaussian integers $\mathbb{Z}[i]$ and the ring of Eisenstein integers $\mathbb{Z}[\omega]$.

Modules over PID and structure theorem

Again, let S be a commutative ring with identity 1. An S-module M over S is an Abelian group $(M, +)$ under a binary operation $+$, together with a function $\mathscr{F} : S \times M \longmapsto M$ which satisfies the same conditions as those for vector space. Note that modules over a field are the same as vector spaces. An S-submodule of M is a subgroup N of M which is closed under the action of ring elements, and hence the submodule N forms also an S-module under the restricted operations.

An S-module is said to be finitely generated (*f.g.*) if M has a finite basis $\{m_1, \cdots, m_n\}$ such that $\sum_i Rm_i = M$.

The annihilator of an element $m \in M$ is the set of elements $s \in S$ such that $sm = 0$. The annihilator of M is the elements $s \in S$ such that $\{sm = 0 | \forall m \in M\}$, denoted by $\mathrm{Ann}_S(M) = \bigcap\{\mathrm{Ann}_S(m) | m \in M\}$. If M is a free S-module, then $\mathrm{Ann}_S(M) = \langle 0 \rangle$.

If M is annihilated by ideal $\mathfrak{I}$ of S, we can make M into a quotient S-module M/N by defining an action on M satisfying,

$$m(s + \mathfrak{I}) = ms, \quad \forall m \in M$$

The torsion submodule M_{Tor} of M is defined by:

$$M_{\mathrm{Tor}} = \{m \in M : \mathrm{Ann}_S(m) \neq \{0\}\}$$

A torsion free module is trivial.

Let M and N be two S-modules. An S-module homomorphism is a map $\phi : M \longmapsto N$, which respects the S-module structures of M and N, i.e.,

$$\phi(s_1 m_1 + s_2 m_2) = s_1 \phi(m_1) \odot s_2 \phi(m_2)$$

$\forall s_1, s_2 \in S, \forall m_1, m_2 \in M$. An S-module homomorphism $\phi : M \longmapsto N$ is called an S-module isomorphism if it is both injective and surjective, which is denoted by $M \cong N$. The kernel of ϕ denotes the elements in M which makes the image of ϕ equal to zero.

3.2.3 Multilevel lattice network coding

Algebraic approach for multilevel structure

Briefly if there is a matrix $\mathbf{G}_\Lambda \in \mathbb{C}^{n' \times n}$, $n' \leq n$ such that all its n' row vectors

$$\mathbf{g}_{\Lambda,1}, \cdots, \mathbf{g}_{\Lambda,n'} \in \mathbb{C}^n$$

are linearly independent, the set of all S-linear combinations of $\mathbf{g}_{\Lambda,1}, \cdots, \mathbf{g}_{\Lambda,n'}$ forms an S-lattice $\Lambda \in \mathbb{C}^n$, written by, $\Lambda = \{\mathbf{s}\mathbf{G}_\Lambda : \mathbf{s} \in S^{n'}\}$, where $\mathbf{G}_\Lambda$ is called the lattice generator.

Following the explanation in Section 3.2.2, an n-dimensional S-lattice is precisely an S-module over PID, and similarly the sublattice Λ' in Λ forms a S-submodule. The partition of the S-lattice, denoted by Λ/Λ' represents $|\Lambda : \Lambda'| < \infty$ (the index of Λ') equivalence classes. Assume S is a PID, we have the following theorem.

Theorem 3.1. *Let Λ and Λ' be S-lattices and S-sublattices, $\Lambda' \subseteq \Lambda$, $|\Lambda : \Lambda'| < \infty$ such that Λ/Λ' has nonzero annihilators. Then Λ/Λ' is the direct sum of a finite number of quotient sublattices,*

$$\Lambda/\Lambda' = \Lambda_{p_1}/\Lambda'_{p_1} \oplus \Lambda_{p_2}/\Lambda'_{p_2} \oplus \cdots \oplus \Lambda_{p_m}/\Lambda'_{p_m} \tag{3.29}$$

where $\Lambda_{p_i}/\Lambda'_{p_i} \triangleq \{\lambda \in \Lambda/\Lambda' : p_i^{\gamma}\lambda = 0\}$ for some $\gamma \geqslant 1$, and every p_i, $i = 1, 2, \cdots, m$ is a distinct prime over S.

Proof: The quotient S-lattice Λ/Λ' has non-zero annihilators; this implies that Λ/Λ' forms a $f.g.$ torsion module. Let $\lambda \in \Lambda/\Lambda'$ and suppose that $\mathrm{Ann}_S(\Lambda/\Lambda') = Sa$, where $a \in S$ and $a \neq 0$ (the property of the torsion module). Since S is also a unique factorisation domain, so $a = p_1^{\gamma_1}p_2^{\gamma_2}\cdots, p_m^{\gamma_m}$. We now write $a_i = a/p_i^{\gamma_i}$ which is the product of irreducible factors that are relatively prime to p_i. There must exist $s_1, s_2, \cdots, s_m$ in S such $\sum_{i=1}^m s_i a_i = 1$ since $\gcd(a_1, a_2, \cdots, a_m) = 1$; Now we have

$$s_i p_i^{\gamma_i} a_i \lambda = 0 \tag{3.30}$$

since a annihilates Λ/Λ'. Theorem 3.1 states that the sublattice $\Lambda_{p_i}/\Lambda'_{p_i}$ has to satisfy the condition $p_i^{\gamma_i}\lambda = 0$ for some γ_is. Hence, if λ is annihilated by some powers of p_i, then $s_i a_i \lambda \in \Lambda_{p_i}/\Lambda'_{p_i}$. Based on the statements above, $\sum_{i=1}^m s_i a_i \lambda = \lambda$, this proves that the S-lattice

$$\lambda \in \Lambda_{p_1}/\Lambda'_{p_1} + \Lambda_{p_2}/\Lambda'_{p_2} + \cdots + \Lambda_{p_m}/\Lambda'_{p_m} \tag{3.31}$$

We suppose $\lambda_i \in \Lambda_{p_i}/\Lambda'_{p_i}$ and $\sum_{i=1}^m \lambda_i = 0$. Then

$$a_i \sum_{j=1}^{m} \lambda_j = a_i \lambda_i = 0 \tag{3.32}$$

where (3.32) follows from the fact that $a_i\lambda_j = 0$ for $i \neq j$. Since a_i is non-zero, λ_i has to be zero. Based on the same proof, we can conclude that $\{\lambda_i = 0 | \forall i = 1, 2, \cdots, m\}$ provided that $\sum_{i=1}^m \lambda_i = 0$. This suggests that every $\lambda \in \Lambda/\Lambda'$ can be uniquely expressed as the summation of the primary sublattice λ_i, $i = 1, 2, \cdots, m$. It implies that there exists a map π:

$$\begin{aligned} &\Lambda_{p_1}/\Lambda'_{p_1} \oplus \Lambda_{p_2}/\Lambda'_{p_2} \oplus \cdots \oplus \Lambda_{p_m}/\Lambda'_{p_m} \\ &\longmapsto \Lambda_{p_1}/\Lambda'_{p_1} + \Lambda_{p_2}/\Lambda'_{p_2} + \cdots + \Lambda_{p_m}/\Lambda'_{p_m} \end{aligned} \tag{3.33}$$

defined by

$$\begin{aligned}&\pi(\Lambda_{p_1}/\Lambda'_{p_1} \oplus \Lambda_{p_2}/\Lambda'_{p_2} \oplus \cdots \oplus \Lambda_{p_m}/\Lambda'_{p_m})\\ =&\Lambda_{p_1}/\Lambda'_{p_1} + \Lambda_{p_2}/\Lambda'_{p_2} + \cdots + \Lambda_{p_m}/\Lambda'_{p_m}\end{aligned} \tag{3.34}$$

which is an S-module isomorphism, and also that

$$\Lambda_{p_i}/\Lambda'_{p_i} \cap \sum_{j=1, j\neq i}^{m} \Lambda_{p_j}/\Lambda'_{p_j} = \mathbf{0} \tag{3.35}$$

This proves that the sum $\sum_{j=1}^{m} \Lambda_{p_j}/\Lambda'_{p_j}$ is direct, and hence the map π is an identity map which belongs to automorphism. Theorem 3.1 is thus proved. ■

Theorem 3.1 proves that Λ/Λ' can be decomposed into the direct sum of m sublattices $\Lambda_{p_i}/\Lambda'_{p_i}$ (the primary sublattices) which itself forms a new lattice system. Hence, Λ/Λ' can be regarded as an m layer quotient lattice.

Theorem 3.2. *Every primary sublattice $\Lambda_{p_i}/\Lambda'_{p_i}$ is isomorphic to a direct sum of cyclic p_i-torsion modules:*

$$\Lambda_{p_i}/\Lambda'_{p_i} \cong S/\langle p_i^{\theta_1}\rangle \oplus S/\langle p_i^{\theta_2}\rangle \oplus \cdots \oplus S/\langle p_i^{\theta_t}\rangle \tag{3.36}$$

for some integers $1 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_t$ which are uniquely determined by $\Lambda_{p_i}/\Lambda'_{p_i}$.

Proof: Theorem 3.1 implies that $\Lambda_{p_i}/\Lambda'_{p_i}$ is an $f.g.$ torsion module. Here we write $M_{p_i} = \Lambda_{p_i}/\Lambda'_{p_i}$; let $x_1, \cdots, x_f$ be generators for M_{p_i} where f is minimal. This means that M_{p_i}/Sx_1 is generated by $f-1$ elements, and hence it is a direct sum of $\leq f-1$ cyclic torsion modules. Thus, if $\lambda \in M_{p_i}$ satisfies $p_i^{\theta_t}\lambda = 0$ and $p_i^{\theta_t - 1}\lambda \neq 0$, then $M_{p_i} = S\lambda \oplus N \cong (S/\langle p_i^{\theta_t}\rangle)^k \oplus N$, where N is the submodule which is not annihilated by $p_i^{\theta_t}$ although it is annihilated by some other powers of p_i. Given this, we have $M_{p_i}/pM_{p_i} \cong (S/\langle p\rangle)^k \oplus N/pN$ (if $S/\langle p\rangle$ exists), the dimension of the second term, $\dim(N/pN) = \dim(M_{p_i}/pM_{p_i}) - k$. It is clear that the dimension is reduced when the power of p_i increases, until the process ends. This proves that $\Lambda_{p_i}/\Lambda'_{p_i}$ consists of a direct sum of cyclic torsion modules, and hence has to be isomorphic to a direct sum of quotient rings over some powers of p_i. This proves the existence of (3.36).

There exists a chain $0 = p_i^{\theta_t}M_{p_i} \subset \cdots \subset p_i^2 M_{p_i} \subset p_i M_{p_i} \subset M_{p_i}$. Consider that $p_i^{\theta-1}M_{p_i} \cong p_i^{\theta-1}S/p_i^{\theta_1}S \oplus \cdots \oplus p_i^{\theta-1}S/p_i^{\theta_t}S \cong S/\langle p_i^{\theta_t-\theta+1}\rangle \oplus \cdots \oplus S/\langle p_i^{\theta_s-\theta+1}\rangle$. This follows from the third isomorphism theorem and that for those $\theta \geq \theta_i$, $p_i^{\theta}(S/\langle p_i^{\theta_i}\rangle) = 0$. Hence, $p_i^{\theta-1}M_{p_i}/p_i^{\theta}M_{p_i} \cong (S/\langle p_i\rangle)^k$ forms a vector space over $S/\langle p_i\rangle$ where k is the number of elementary divisors p_i^{α} with $\alpha \geq \theta$. Thus, $\dim(p_i^{\theta-1}M_{p_i}/p_i^{\theta}M_{p_i})$ is the number of elements in (3.36) whose $\theta_i \geq \theta$. This proves that the dimension of $p_i^{\theta-1}M_{p_i}/p_i^{\theta}M_{p_i}$ is invariant with M_{p_i}, and the number of summands in a particular form $S/\langle p_i^{\theta_i}\rangle$ is uniquely determined by M_{p_i}. This proves the uniqueness of (3.36). ■

Theorem 3.2 implies that the quotient primary S-sublattice system $\Lambda_{p_i}/\Lambda'_{p_i}$ is isomorphic to a cyclic p_i-torsion module. The right-hand side of (3.36) can be viewed as the message space of $\Lambda_{p_i}/\Lambda'_{p_i}$ which is detailed in Lemma 3.1.

Lemma 3.1. *There exists a map:*

$$\phi_i : \Lambda_{p_i} \longmapsto \bigoplus_j S/\langle p_i^{\theta_j} \rangle \tag{3.37}$$

which is a surjective S-module homomorphism with kernel $\mathcal{K}(\phi_i) = \Lambda'_{p_i}$. To ease the abstract representation, we consider $\Lambda'_{p_i} = \Lambda'$ in the sequel. Thus, $\mathcal{K}(\phi_i) = \Lambda'$ for $i = 1, 2, \cdots, m$. If the message space is taken as the canonical decomposition of (3.36), i.e. $\mathbf{w}^i = \bigoplus_j S/\langle p_i^{\theta_j} \rangle$, there exists a surjective homomorphism ϕ and also an injective map $\tilde{\phi} : (\mathbf{w}^1, \cdots, \mathbf{w}^m) \longmapsto \Lambda$ such that

$$\phi(\tilde{\phi}(\mathbf{w}^1 \oplus \cdots \oplus \mathbf{w}^m)) = \mathbf{w}^1 \oplus \cdots \oplus \mathbf{w}^m \tag{3.38}$$

Proof: The statement of (3.37) follows immediately from Theorem 3.2 and the first isomorphism theorem. The statement of (3.38) follows from Theorem 3.1, Theorem 3.2 and the first isomorphism theorem. ■

Lemma 3.2. *The generator matrix of the S-sublattice Λ_{p_i} at the i^{th} layer can be expressed in the form of:*

$$\mathbf{G}_{\Lambda_{p_i}} = \begin{bmatrix} \text{Diag}(\underbrace{\mathbf{p}_1^{\theta_1} \cdots \mathbf{p}_{i-1}^{\theta_{i-1}}, \mathbf{I}_t, \mathbf{p}_{i+1}^{\theta_{i+1}} \cdots \mathbf{p}_m^{\theta_m}}_{k}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{bmatrix} \mathbf{G}_\Lambda \tag{3.39}$$

and

$$\phi_i(\mathbf{w}\mathbf{G}_{\Lambda_{p_i}}) = \left(w^{i,1} + \langle p_i^{\theta_1} \rangle, \cdots, w^{i,t} + \langle p_i^{\theta_t} \rangle\right) \tag{3.40}$$

where $w^{i,t} \in S/\langle p_i^{\theta_t} \rangle$ and $\mathbf{w} \in \mathbf{w}^1 \oplus \cdots \oplus \mathbf{w}^m$. $\mathbf{G}_\Lambda$ is the generator matrix of the fine lattice Λ, $\mathbf{p}_j^{\theta_j}$, $j = 1, 2, \cdots, m$ is a vector, with all elements being the same elementary divisor $p_j^{\theta_j}$ over S, and $t = \dim(\Lambda_{p_i}/\Lambda')$.

Proof: Every matrix over a PID has to have a Smith normal form (SNF) with unique invariant factors up to multiplication by units. This complies with the structure theorem of modules over PID in invariant factor form. Hence, there exists an *equivalent* SNF matrix $\mathbf{M}_{\text{SNF}}$ such that $\mathbf{M}_{\text{SNF}}\mathbf{G}_{\Lambda_{p_i}}$ is the generator matrix of the lattice $\tilde{\Lambda}'$ which is isomorphic to the kernel $\mathcal{K}(\phi_i) = \Lambda'$. Based on the theorems mentioned above and the fact that the invariant factors are uniquely determined, the invariant factors in $\mathbf{M}_{\text{SNF}}$ has to be some powers of p_i which naturally satisfies the divisibility relations, and we claim that, now $\tilde{\Lambda}' = \Lambda'$. The statement of (3.40) follows from Lemma 3.1. ■

Lemma 3.2 shows a way to produce the quotient S-sublattice of each layer defined in Theorem 3.1. Λ_{p_i}/Λ' forms an independent lattice system, and the direct sum of all Λ_{p_i}/Λ', $i = 1, 2, \cdots, m$ is equal to Λ/Λ'.

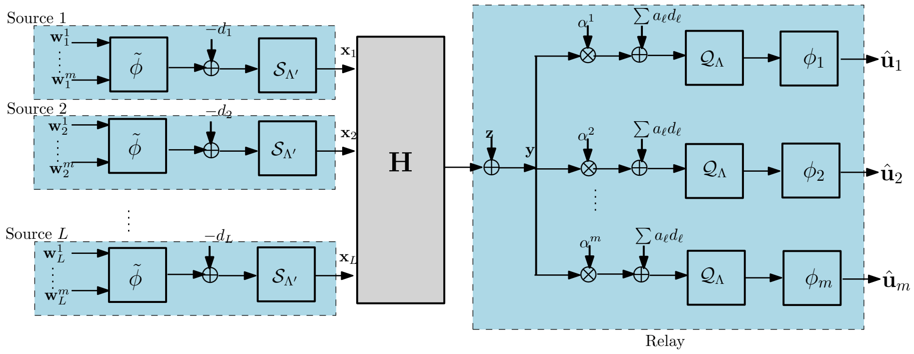


Figure 3.10: System diagram of the multilevel lattice network coding and multistage decoding. The right-hand side of $\mathbf{H}$ represents the decoding for a single relay.

Construction of multilevel lattice network coding

Based on the theorems developed in Section 3.2.3, we show in this subsection the detailed description of the MLNC scheme, and a way of multilevel network decoding (named layered integer forcing), which provides an efficient way of decoding the linear combination of the multi-source messages with greatly reduced complexity.

Traditional approach. Lemma 3.1, Theorem 3.1 and Theorem 3.2 imply that the message space with large cardinality may be expressed as a set of smaller message spaces over the hybrid finite field and finite chain ring. Figure 3.10 depicts a multilevel lattice network coding architecture, with L sources and a single relay. The encoder $\mathcal{E}_\ell$ at the ℓ^{th} source maps the original message $\mathbf{w}_\ell = \mathbf{w}_\ell^1 \oplus \cdots \oplus \mathbf{w}_\ell^m$ to a fine lattice point Λ (assuming n-dimension) via the injective map $\tilde{\phi}$ defined in Lemma 3.1. Then we add a dither $\boldsymbol{d}_\ell \in \mathbb{C}^n$ which is uniformly distributed over the fundamental Voronoi region $\mathcal{V}_{\Lambda'}$ of Λ'. The dithered lattices pass through a nested shaping operator in order to restrain the power consumption. This operation is performed via the sublattice quantisation:

$$\lambda'_\ell = \mathcal{Q}_{\Lambda'}(\tilde{\phi}(\mathbf{w}_\ell^1 \oplus \cdots \oplus \mathbf{w}_\ell^m) + \boldsymbol{d}_\ell) \tag{3.41}$$

where $\lambda'_\ell \in \Lambda'$, and $\mathcal{Q}_{\Lambda'}(\,\cdot\,) : \mathbb{C}^n \longmapsto \Lambda'$ is a coarse lattice quantiser. The output of the ℓ^{th} source is given by:

$$\begin{aligned} \mathbf{x}_\ell &= \mathcal{E}_\ell(\mathbf{w}_\ell^1 \oplus \cdots \oplus \mathbf{w}_\ell^m) \\ &= \tilde{\phi}(\mathbf{w}_\ell^1 \oplus \cdots \oplus \mathbf{w}_\ell^m) + \boldsymbol{d}_\ell - \lambda'_\ell \end{aligned} \tag{3.42}$$

Note that $\mathbf{x}_\ell$ is uniformly distributed over $\mathcal{V}_{\Lambda'}$ due to the effect of the dither. The average power of the transmitted signal $\mathbf{x}_\ell$ is given by:

$$P = \frac{1}{n\mathrm{Vol}(\mathcal{V}_{\Lambda'})} \int_{\mathcal{V}_{\Lambda'}} \parallel \mathbf{x}_\ell \parallel^2 \mathrm{d}\mathbf{x}_\ell \tag{3.43}$$

which is the second moment per dimension of $\mathbf{x}_\ell$ over $\mathcal{V}_{\Lambda'}$. The message space at each source consists of a direct sum of m small message spaces (assuming there are m levels) over different finite fields or chain rings. The encoder $\mathcal{E}_\ell$ constructs a one-to-one relation between the message space and the coset system Λ/Λ'.

At the relay, given the received signals $\mathbf{y}$ and an S-integer vector $\tilde{\mathbf{a}} = [\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_L]^T \in S^L$, the decoder aims at computing a new lattice point which is regarded as an S-linear combination of transmitted lattice points from all sources. The homomorphism designed for the coset system will be used for decoding the lattice point to a linear combination of the original messages. We assume in this paper that the fading coefficients $\mathbf{h} = [h_1, h_2, \cdots, h_L]$, and dithers are perfectly known at the relay. The decoder can be described, generally, by:

$$\mathcal{D} : (\mathbb{C}^n, \mathbb{C}^L, S^L, \mathbb{C}, \mathbb{C}^{n\times L}) \longmapsto W \tag{3.44}$$

Thus, the output of $\mathcal{D}(\mathbf{y}|\mathbf{h}, \tilde{\mathbf{a}}, \alpha, \mathbf{d})$ is the estimates of the linear combination of the original messages of each source. Here α is a scaling factor [22] which maximises the computation rate. Note that the aforementioned decoder (3.44) may vary according to the specific problem. There may be additional information available to the decoder, and the decoder may also have extra outputs. However, basically the core idea for the decoding remains the same. Based on the quotient lattice Λ/Λ', we have:

$$\begin{aligned}
\hat{\mathbf{u}} &= \mathcal{D}(\mathbf{y}|\mathbf{h}, \mathbf{a}, \alpha, \mathbf{d}) \\
&\overset{(a)}{=} \phi\Big(\mathcal{Q}_\Lambda\Big(\alpha\mathbf{y} - \sum_{\ell=1}^{L} \tilde{a}_\ell \boldsymbol{d}_\ell\Big)\Big) && (3.45)\\
&\overset{(b)}{=} \phi\Big(\mathcal{Q}_\Lambda\Big(\sum_{\ell=1}^{L} \tilde{a}_\ell(\tilde{\phi}(\mathbf{w}_\ell) - \lambda'_\ell) + \mathbf{n}_{\text{eff}}\Big)\Big) && (3.46)\\
&\overset{(c)}{=} \phi\Big(\sum_{\ell=1}^{L} \tilde{a}_\ell\tilde{\phi}(\mathbf{w}_\ell) + \mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\Big) && (3.47)\\
&\overset{(d)}{=} \bigoplus_{\ell=1}^{L} a_\ell\mathbf{w}_\ell \boxplus \phi\left(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\right) && (3.48)
\end{aligned}$$

where (a) follows from the fact that we expect to quantise a set of scaled received signals which are subtracted from the corresponding dithers. (b) follows from the manipulation of:

$$\alpha\mathbf{y} = \sum_{\ell=1}^{L} \tilde{a}_\ell\mathbf{x}_\ell + \sum_{\ell=1}^{L} \tilde{a}_\ell\boldsymbol{d}_\ell + \overbrace{\sum_{\ell=1}^{L}(\alpha h_\ell - \tilde{a}_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}}^{\mathbf{n}_{\text{eff}}} \tag{3.49}$$

(c) follows from the definition of the lattice quantiser, and (d) follows from the properties of a surjective module homomorphism, and also Lemma 3.1. Note that here $\phi(\tilde{a}_\ell) = a_\ell \in \mathbf{w}^1 \oplus \cdots \oplus \mathbf{w}^m$.

Equations (3.45) – (3.48) reveal the decoding operations for the traditional lattice-based WPLNC. We are able to decode a linear combination of messages $\bigoplus_{\ell=1}^{L} a_\ell \mathbf{w}_\ell$ over all sources without errors provided that $\phi\left(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\right) = 0$. Thus, the successful decoding is guaranteed iff the effective noise is quantised to the kernel of ϕ, $\mathcal{K}(\phi)$.

The problems left unsolved are: 1. How to exploit rich ring features in order to make it practically applicable in lattice-based network coding. 2. When the cardinality (the coset representatives) of Λ/Λ' is large, the complexity of the lattice quantiser becomes unmanageable, which restricts the application of LNC. What is the practical lattice network decoding approach that could greatly relieves the decoding load in LNC. We study a new decoding solution which is specifically designed in terms of MLNC, and which relaxes the two problems mentioned.

Layered integer forcing. The breakthrough of MLNC (based on theorems and lemmas in Section 3.2.3) is that

- The original message space over Λ/Λ' can be decomposed into a direct sum of m smaller message spaces in terms of Λ_{p_i}/Λ', $i = 1, 2, \cdots, m$.
- The relay can decode each layer independently, thus the decoder tries to infer and forward a linear combination of messages of each layer *separately* over the message subspace defined in Theorem 3.2.

Let us recall the traditional decoding operations explained in (3.45) – (3.48). If we are only concerned with the linear combination of a particular layer, the quantisation of the effective noise need not necessarily be the kernel of ϕ. There has to exist other lattice points in Λ/Λ' such that the homomorphism of these points does not interfere with the linear combination of that layer following the aforementioned theorems.

Theorem 3.3. *There exists a quotient S-lattice Λ/Λ'_i with generator matrices $\mathbf{G}_\Lambda$ for Λ, and $\mathbf{G}_{\Lambda'_i}$ for Λ'_i, which satisfies:*

$$\mathbf{G}_{\Lambda'_i} = \begin{bmatrix} \text{Diag}(\underbrace{\mathbf{I}, p_i^{\theta_1}, \cdots, p_i^{\theta_t}, \mathbf{I}}_{k}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{bmatrix} \mathbf{G}_\Lambda \tag{3.50}$$

and there is a surjective S-module homomorphism φ_i:

$$\varphi_i : \Lambda \longmapsto S/\langle p_i^{\theta_1}\rangle \oplus S/\langle p_i^{\theta_2}\rangle \oplus \cdots \oplus S/\langle p_i^{\theta_t}\rangle \tag{3.51}$$

whose kernel $\mathcal{K}(\varphi_i) = \Lambda'_i$. The quotient S-lattice Λ/Λ'_i is isomorphic to the direct sum of cyclic modules:

$$\Lambda/\Lambda'_i \cong S/\langle p_i^{\theta_1}\rangle \oplus S/\langle p_i^{\theta_2}\rangle \oplus \cdots \oplus S/\langle p_i^{\theta_t}\rangle \tag{3.52}$$

Proof: By applying the equivalent SNF explained in the proof of Lemma 3.2, we can prove (3.50) in similar way. Now we begin the proof of (3.51). The sublattice Λ_i' can be written as:

$$\Lambda_i' = \{\mathbf{w}\mathbf{G}_\Lambda : \mathbf{w}^i \in \langle p_i^{\theta_1}\rangle \oplus \cdots \oplus \langle p_i^{\theta_t}\rangle\}$$

in terms of the generator matrix for Λ_i' in (3.50). It is clear that:

$$\varphi_i(\mathbf{w}\mathbf{G}_\Lambda) = 0 \;\; \textbf{iff} \;\; \mathbf{w}^i \in \langle p_i^{\theta_1}\rangle \oplus \cdots \oplus \langle p_i^{\theta_t}\rangle$$

and hence, the kernel of φ_i, $\mathcal{K}(\varphi_i)$ has to be Λ_i'. It is also obvious from (3.51) that φ_i is indeed surjective and S-linear. The proof of (3.52) follows immediately from the first isomorphism theorem. ■

Note that although both Λ_{p_i}/Λ' and Λ/Λ_i' are isomorphic to $S/\langle p_i^{\theta_1}\rangle \oplus S/\langle p_i^{\theta_2}\rangle \oplus \cdots \oplus S/\langle p_i^{\theta_t}\rangle$, *they belong to different coset systems.* Λ_{p_i}/Λ' is related to the construction of lattices that have multilevel structure, whereas Λ/Λ_i' is related to the decoding issues, i.e. LIF.

Theorem 3.3 defines a new sublattice Λ_i' which plays a key role in decoding MLNC, as it is the kernel of the quotient S-lattice that possesses a surjective homomorphism φ_i for the i^{th} layer. Hence, it is possible to decode an S-linear combination of fine lattice points to an S-linear combination of the original messages of the i^{th} layer. This is explained in Lemma 3.3.

Lemma 3.3. *Given the embedding injective map $\tilde{\phi} : (\mathbf{w}^1, \cdots, \mathbf{w}^m) \longmapsto \Lambda$, there exists a surjective S-module homomorphism φ_i, $i = 1, 2, \cdots, m$, defined in (3.51), satisfying:*

$$\varphi_i\big(\tilde{\phi}(\mathbf{w}^1 \oplus \cdots \oplus \mathbf{w}^m)\big) = \begin{cases} \mathbf{w}^i, \mathbf{w}^i \notin \langle p_i^{\theta_1}\rangle \oplus \cdots \oplus \langle p_i^{\theta_t}\rangle \\ 0, \;\; \mathbf{w}^i \in \langle p_i^{\theta_1}\rangle \oplus \cdots \oplus \langle p_i^{\theta_t}\rangle \end{cases} \tag{3.53}$$

Proof: The injective mapping $\tilde{\phi}$ is an inverse operation of the homomorphism ϕ defined in terms of the quotient S-lattice Λ/Λ', which maps the messages into a lattice point Λ, as explained in Lemma 3.1. Following the second statement of Theorem 3.3, the S-module homomorphism φ_i of the i^{th} layer indeed maps the lattice point Λ to the message subspace. When $\mathbf{w}^i \in \langle p_i^{\theta_1}\rangle \oplus \cdots \oplus \langle p_i^{\theta_t}\rangle$, $\tilde{\phi}(\mathbf{w}^1 \oplus \cdots \oplus \mathbf{w}^m)$ maps to the lattice point Λ_i' which is the kernel of φ_i. Hence, according to (3.50), the linear labelling of the new coset system in Λ/Λ_i' coincides with the labelling of the i^{th} layer of $\tilde{\phi}$. This proves Lemma 3.3. ■

Based on Lemma 3.3, it is now possible to decode the linear combination of the messages of each layer separately and independently. Assuming the messages at the i^{th} layer is of interest, the relay computes:

$$\hat{\mathbf{u}}^i = \mathcal{D}^i(\mathbf{y}|\mathbf{h}, \mathbf{a}^i, \alpha^i, \mathbf{d}) \tag{3.54}$$

$$= \varphi_i\bigg(\mathcal{Q}_\Lambda\Big(\alpha^i\mathbf{y} - \sum_{\ell=1}^{L} \tilde{a}_\ell^i \boldsymbol{d}_\ell\Big)\bigg) \tag{3.55}$$

where

$$\mathcal{D}^i : (\mathbb{C}^n, \mathbb{C}^L, S^L, \mathbb{C}, \mathbb{C}^{n\times L}) \longmapsto W^i \tag{3.56}$$

and $\alpha^i \in \mathbb{C}$ and $\mathbf{a}^i$ are scaling parameter and S-integer coefficients of the i^{th} layer, respectively, which are determined by some optimisation criterion in terms of the quotient S-lattice Λ/Λ'_i.

Theorem 3.3 and Lemma 3.3 lay the foundation of the layered integer forcing. The linear combination of $\hat{\mathbf{u}}^i$ can be recovered in terms of LIF by:

$$\begin{aligned}
\hat{\mathbf{u}}^i &\overset{(d)}{=} \varphi_i\Big(\mathcal{Q}_\Lambda\Big(\sum_{\ell=1}^{L} \tilde{a}^i_\ell\big(\tilde{\phi}(\mathbf{w}^1_\ell \oplus \cdots \oplus \mathbf{w}^m_\ell) - \lambda'_\ell\big) + \mathbf{n}_{\text{eff}}\Big)\Big)\\
&\overset{(e)}{=} \varphi_i\Big(\sum_{\ell=1}^{L} \tilde{a}^i_\ell\tilde{\phi}(\mathbf{w}^1_\ell \oplus \cdots \oplus \mathbf{w}^m_\ell) - \lambda'_\ell - \lambda'_{i,\ell} + \mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\Big)\\
&\overset{(f)}{=} \varphi_i\Big(\sum_{\ell=1}^{L} \tilde{a}^i_\ell\tilde{\phi}(\mathbf{w}^1_\ell \oplus \cdots \oplus \mathbf{w}^m_\ell)\Big) \boxplus \varphi_i\Big(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\Big)\\
&\overset{(g)}{=} \bigoplus_{\ell=1}^{L} a^i_\ell \mathbf{w}^i_\ell \boxplus \varphi_i\Big(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\Big)
\end{aligned} \tag{3.57}$$

where (d) follows from (3.42) and basic arithmetic manipulations; (e) follows from the definition of the lattice quantiser $\mathcal{Q}_\Lambda$, and also the S-linear combination of the lattice points is restricted in $\mathcal{V}_{\Lambda'_i}$; (f) follows from the property of a surjective S-module homomorphism, and also the fact that $\lambda' \subseteq \lambda'_i$ and $\mathcal{K}(\varphi_i) = \lambda'_i$. (g) follows from Lemma 3.3, and note that $\varphi_i(\tilde{a}^i_\ell) = a^i_\ell \in W^i$.

Lemma 3.4. *The linear combination of the messages at the i^{th} layer $\hat{\mathbf{u}}^i = \bigoplus_{\ell=1}^{L} a^i_\ell \mathbf{w}^i_\ell$ can be recovered iff $\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}}) \in \Lambda'_i$. Thus,* $\Pr(\hat{\mathbf{u}}^i \neq \mathbf{u}^i) = \Pr(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}}) \notin \Lambda'_i)$.

Proof: Following (3.57), it is clear that $\hat{\mathbf{u}}^i = \bigoplus_{\ell=1}^{L} a^i_\ell \mathbf{w}^i_\ell$ can be decoded correctly iff $\varphi_i\Big(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})\Big) = 0$. According to Theorem 3.3, the kernel of φ_i, $\mathcal{K}(\varphi_i) = \Lambda'_i$, thus the quantisation of the effective noise $\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}}) \in \Lambda'_i$. This proves Lemma 3.4. ■

Lemma 3.4 reveals that the lattice Λ'_i defined in Theorem 3.3 plays a key role in decoding the messages of the i^{th} layer.

The message space of the traditional CaF scheme is determined by the size of the lattice partition. Hence, to increase the network throughput, the sublattice Λ' needs to be more sparse in order to allow the messages to be over a larger field or commutative ring (LNC). In this case, the decoding complexity is normally unaffordable.

One example is associated with a group of lattice codes directly designed in the Euclidean space, e.g. complex low density lattice codes (CLDLC). It has prohibitive computational complexity when the cardinality of the quotient lattice is too large, since the decoding metrics are continuous functions (a mixture of multiple probability density functions), and the periodic extension that occurs at the variable nodes [59] runs over a large S-integer set, which seriously increases the overall computational costs over the iterative parametric belief propagation decoding, even if the Gaussian mixture reduction algorithm is employed.

The S-lattices can also be constructed through the existing channel codes based on some lattice construction approaches, e.g. Construction A or D. However, the decoding complexity of the channel codes over a large algebraic field increases rapidly, e.g. a small increase of the memory for convolutional codes gives rise to an exponential increase in the number of trellis states, making the codes eventually undecodable. When the cardinality of the quotient lattices become larger, the decoding complexity for convolutional codes with even small memory is unmanageable, but the performance is still very poor.

MLNC together with LIF provides a realistic solution to this problem. Being supported by the theorems and lemmas in Sections 3.2.3 and 3.2.3, the quotient S-lattice having large cardinality can be decomposed into some primary quotient S-sublattices which have smaller cardinalities. Each primary quotient sublattice forms a layer, and determines the message subspace over this layer. With the aid of the lattices Λ'_i, we can perform multilevel lattice decoding at the relay, where the linear combination of the messages of all sources at each layer can be independently recovered over the message subspace. In this case, the overall computational loads are greatly relaxed.

LNC [49] shows the possibility of ring-based linear network coding, extending the traditional linear network coding defined over the finite field to a more general notion. Furthermore, MLNC leads to a practically feasible encoding and decoding design approach for lattice network coding over commutative rings, thus, with greatly reduced decoding complexity. MLNC inherently gives an appealing solution for this since now we are able to construct multiple layers based on the decomposition theory mentioned above, with each layer operating over a finite field or chain ring in a new coset system. Note that the elements in a finite chain ring can be uniquely represented by $\nu + 1$ elements over a fixed residue field where ν is the nilpotency index of this finite chain ring. We will introduce this in the subsequent sections.

Achievable rates and probability of error

As discussed in Section 3.2.3, the message of the i^{th} layer corresponds to the decomposed quotient S-sublattice Λ_{p_i}/Λ', which should be decoded separately at each layer, based on a new S-lattice partition Λ/Λ'_i. Suppose that each layer is given an S-integer coefficient vector $\mathbf{a}^i \in S^L$, and $\mathbf{A} = [\mathbf{a}^1|\mathbf{a}^2|\cdots|\mathbf{a}^m] \in S^{L\times m}$, we can obtain the achievable rate following Nazer and Gastpar's method, under the assumption of that S is Gaussian integers $\mathbb{Z}[i]$

Theorem 3.4. *Given channel fading vector* $\mathbf{h} \in \mathbb{C}^L$*, non-zeros* S*-integer coefficient matrix* $\mathbf{A} \notin \{\mathbf{0}\}$*, and the message subspace* $W^i = \left(\mathbb{Z}[i]/\langle p_i \rangle\right)^k$*, the probability of decoding error* $\Pr(\hat{\mathbf{u}}^i \neq \mathbf{u}^i | \mathbf{h}, \mathbf{A})$ *can be arbitrarily small if the overall message rate* $\mathcal{R}$ *satisfies:*

$$\mathcal{R} < \mathcal{R}(\mathbf{h}, \mathbf{A}) = \sum_{i}^{m} \log_2 \left(\left(\parallel \mathbf{a}^i \parallel^2 - \frac{P^i |\mathbf{h}^\dagger \mathbf{a}^i|^2}{1 + P^i \parallel \mathbf{h} \parallel^2} \right)^{-1} \right) \tag{3.58}$$

for sufficiently large lattice dimension n *and prime factor* p_i*.* $P^{(i)}$ *is defined by*

$$P^{(i)} = \frac{1}{n\mathrm{Vol}(\mathcal{V}_{\Lambda/\Lambda_i'})} \int_{\mathcal{V}_{\Lambda/\Lambda_i'}} \parallel \mathbf{x}_\ell \parallel^2 \mathrm{d}\mathbf{x}_\ell \tag{3.59}$$

Proof: Suppose there are m layers, we can construct a quotient $\mathbb{Z}[i]$-lattice Λ/Λ_i' which is isomorphic to the message subspace W^i. The computation rate of each layer follows from Nazer and Gastpar's method in [22]. Since each layer is decoded independently, the sum of computation rate of all layers is the overall achievable rate. ■

Recall Lemma 3.4, the error probability of decoding a linear combination $\mathbf{u}$ in terms of $\mathbf{a}^i$ for the i^{th} level is equal to the probability of $\Pr(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}}) \notin \Lambda_i')$. The union bound of the error probability for MLNC is given by:

Theorem 3.5. *Given* $\mathbf{h} \in \mathbb{C}^L$*, non-zeros* S*-integer coefficient matrix* $\mathbf{A} \notin \{\mathbf{0}\}$*, and the optimal scaling factor* α_{opt}*, the union bound of the error probability in decoding the linear combinations of all levels in MLNC is given by:*

$$\begin{aligned} &\Pr\left(\hat{\mathbf{u}} \neq \mathbf{u} | \mathbf{h}, \mathbf{A}, \alpha_{\text{opt}} \right) \\ =&\boldsymbol{E}_{p(\mathcal{Z})}\left[\Pr(\hat{\mathbf{u}}^i \neq \mathbf{u}^i | \mathbf{h}, \mathbf{A}, \alpha_{\text{opt}}) \right] \\ \lessapprox&\boldsymbol{E}_{p(\mathcal{Z})}\left[\mathcal{N}(\Lambda/\Lambda_i') \exp\left(\frac{-d^2(\Lambda/\Lambda_i')}{4(N_0|\alpha_{\text{opt}}|^2 + P^i||\alpha_{\text{opt}}\mathbf{h} - \mathbf{a}^i||^2)} \right) \right] \end{aligned} \tag{3.60}$$

where $\mathcal{Z}$ *is a random variable with its outcomes taking on* $\{r = \frac{\dim(\mathbf{u}^i)}{\dim(\mathbf{u})} | i = 1, 2, \cdots, m\}$.

Proof: At the i^{th} layer, the decoding operates over the lattice partition of $\Lambda/\Lambda_i' = \{\Lambda \setminus \Lambda_i'\} \cup \{\mathbf{0}\}$. $d(\Lambda/\Lambda_i')$ is the minimum inter-coset distance of the lattice partition Λ/Λ_i' defined by

$$d(\Lambda/\Lambda_i') = \min ||\boldsymbol{\vartheta}_1 - \boldsymbol{\vartheta}_2||^2, \boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2 \in \Lambda/\Lambda_i', \boldsymbol{\vartheta}_1 \neq \boldsymbol{\vartheta}_2$$

We denote $\mathcal{N}(\Lambda/\Lambda_i')$ as the number of $d(\Lambda/\Lambda_i')$ in the i^{th} layer coset system. Following the steps in [49], we can prove that the probability error of effective noise quantisation is bounded by the probability of effective noise which is not within the Voronoi region $\mathcal{V}_{\mathbf{0}}$:

$$\Pr\Big(\mathcal{Q}_{\Lambda}(\mathbf{n}_{\text{eff}}) \notin \Lambda_i' | \mathbf{h}, \mathbf{a}^i, \alpha_{\text{opt}}\Big) \leq \Pr\Big(\mathbf{n}_{\text{eff}} \notin \mathcal{V}_{\mathbf{0}}^i | \mathbf{h}, \mathbf{a}^i, \alpha_{\text{opt}}\Big)$$

with

$$\mathcal{V}_{\mathbf{0}}^i = \{\vartheta \in \mathbb{C}^n : ||\vartheta - \mathbf{0}||^2 \leq ||\vartheta - \lambda||^2, \forall \lambda \in \Lambda \setminus \Lambda_i'\}$$

The probability of $\Pr(\mathbf{n}_{\text{eff}} \notin \mathcal{V}_{\mathbf{0}}^i \mathbf{h}, \mathbf{a}^i, \alpha_{\text{opt}})$ is upper bounded by the term within the bracket of (3.60). The proof closely follows from the method given in [49], based on the Chernoff inequality, the moment generating function of a complex Gaussian random vector, and hypercube Voronoi region Λ_i'. We refer to [49] for the detailed proof, and also [60] for the proof under Eisenstein integers. Since each layer decodes the linear combination independently, the average error probability is the expectation of $\Pr(\hat{\mathbf{u}}^i \neq \mathbf{u}^i | \mathbf{h}, \mathbf{A}, \alpha_{\text{opt}})$ over the probability function $p(\mathcal{Z})$. According to Lemma 3.3, we know that the probability $\Pr(\hat{\mathbf{u}}^i \neq \mathbf{u}^i | \mathbf{h}, \mathbf{A}, \alpha_{\text{opt}}) \leq \Pr(\mathbf{n}_{\text{eff}} \notin \mathcal{V}_{\mathbf{0}}^i | \mathbf{h}, \mathbf{a}^i, \alpha_{\text{opt}})$; this gives (3.60). ■

One way to design the homomorphism of Λ/Λ_i' at the i^{th} layer is implied in Theorem 3.5. Thus, $\mathcal{N}(\Lambda/\Lambda_i')$ should be minimised and $d(\Lambda/\Lambda_i')$ is maximised such that the probability of error is as small as possible at the i^{th} layer. It is clear that MLNC has good flexibility in the design of the homomorphism, which determines the achievable rate at some levels.

3.2.4 Elementary divisor construction

In this section, we study a new lattice construction approach, based on the theorems and lemmas developed in Section 3.2.3.

Lemma 3.5. *Let Λ and Λ' be S-lattices and S-sublattices, $\Lambda' \subseteq \Lambda$, $|\Lambda : \Lambda'| < \infty$ such that Λ/Λ' has a nonzero annihilator ϖ which can be uniquely factorised into distinct powers of primes in S, $\varpi = \mathcal{U}(S)p_1^{\gamma_1}p_2^{\gamma_2}\cdots p_m^{\gamma_m}$. Then Λ/Λ' is the direct sum of a finite number of quotient sublattices, $\Lambda_{p_i}/\Lambda' = \{\lambda \in \Lambda/\Lambda' : p_i^{\gamma_i}\lambda = 0\}$, $i = 1, 2, \cdots, m$, and given by,*

$$\Lambda/\Lambda' = \Lambda_{p_1}/\Lambda' \oplus \Lambda_{p_2}/\Lambda' \oplus \cdots \oplus \Lambda_{p_m}/\Lambda' \tag{3.61}$$

Proof: Lemma 3.5 is a special case of Theorem 3.1 where the annihilator of Λ/Λ' is a single S-integer. Therefore, Λ/Λ' has to be the direct sum of some new quotient S-lattices. The annihilator of the Λ_{p_i}/Λ' is precisely $p_i^{\gamma_i}$. ■

Elementary divisor construction

We outline a possible lattice construction solution based on Lemma 3.5 and the statements in Section 3.2.3.

Elementary Divisor Construction (EDC): Let $p_1, p_2, \cdots, p_m$ be some distinct primes in a PID S, and $\varpi = \mathcal{U}(S)p_1^{\gamma_1}p_2^{\gamma_2}\cdots p_m^{\gamma_m}$ is a unique factorisation, $\gamma_i \geq 1$. Let $\mathcal{C}^1, \mathcal{C}^2, \cdots, \mathcal{C}^m$ be m $[n, k_i]$ linear codes over $S/\langle p_1^{\gamma_1}\rangle, S/\langle p_m^{\gamma_m}\rangle, \cdots, S/\langle p_m^{\gamma_m}\rangle$, respectively. The elementary divisor construction lattice is defined by:

$$\Lambda \triangleq \{\lambda \in S^n : \tilde{\sigma}(\lambda) \in \mathcal{C}^1 \oplus \mathcal{C}^2 \oplus \cdots \oplus \mathcal{C}^m\} \tag{3.62}$$

and the sublattice is:

$$\Lambda' \triangleq \{\varpi\lambda : \lambda \in S^n\}$$

where $\tilde{\sigma} : S^n \longmapsto (S/\langle p_1^{\gamma_1}\rangle)^n \oplus (S/\langle p_2^{\gamma_2}\rangle)^n \oplus \cdots \oplus (S/\langle p_m^{\gamma_m}\rangle)^n$ is a natural map obtained by extending the ring homomorphism $\sigma : S \longmapsto S/\langle p_1^{\gamma_1}\rangle \times S/\langle p_2^{\gamma_2}\rangle) \times \cdots \times S/\langle p_m^{\gamma_m}\rangle$ to multiple dimensions. Apparently $\Lambda' \subseteq \Lambda$. The message space under EDC is

$$W = (S/\langle p_1^{\gamma_1}\rangle)^{k_1} \oplus \cdots \oplus (S/\langle p_m^{\gamma_m}\rangle)^{k_m} \tag{3.63}$$

where k_i is the message length of the i^{th} layer which sums up to $k = \sum_{j=1}^{m} k_j$.

The elementary divisor construction is a straightforward extension of Lemma 3.5, which defines a class of lattices constructed by m linear codes, with each operating over either a finite field or a finite chain ring. Hence, the quotient Λ/Λ' has to consist of m primary sublattices Λ_{p_i}/Λ', with each constructed by the i^{th} linear code. The primary sublattices Λ_{p_i} of the i^{th} layer is defined by:

$$\Lambda_{p_i} \triangleq \{\lambda_{p_i} \in \delta_i S : \tilde{\sigma}_i(\lambda_{p_i}) \in C^i)\} \tag{3.64}$$

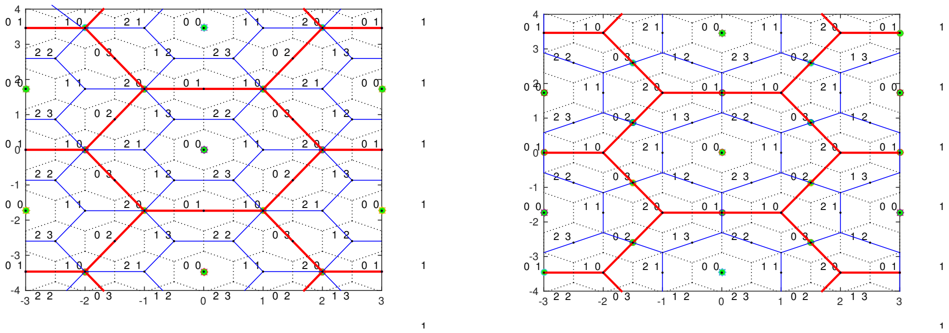
where $\tilde{\sigma}_i$ is a natural map:

$$\tilde{\sigma}_i : (\delta_i S)^n \longmapsto (\delta_i S/p_i^{\gamma_i}\delta_i S)^n \cong (S/\langle p_i^{\gamma_i}\rangle)^{k_i} \tag{3.65}$$

obtained by extending the ring homomorphism $\sigma_i : \delta_i S \longmapsto \delta_i S/\langle p_i^{\gamma_i}\delta_i S\rangle$ to multiple dimensions. The scaling factor $\delta_i = \frac{\varpi}{p_i^{\gamma_i}}$ can be proved in terms of the proof in Theorem 3.1.

We consider three scenarios based on different algebraic fields which the linear codes may belong to.

Scenario 1. Assume that the primary sublattice at each layer is constructed by a linear code over a finite field, thus, $\gamma_1 = \gamma_2 = \cdots = \gamma_m = 1$. Then, $\mathcal{C}^i \in (\delta_i S/\langle p_i\delta_i\rangle)^n$. Since the coarse lattice Λ' is generated by a single element ϖ, Λ/Λ' forms a cyclic torsion



(a) Layer 1

(b) Layer 2

Figure 3.11: Layer structure of a 2-layer EDC lattice. The green points and blue lines represent the primary sublattices and Voronoi region of $\mathcal{V}_{\Lambda_i'}$ for the corresponding layers, respectively. Dotted lines represent the Voronoi region of the fine lattice.

module which allows us to produce the generator matrix of the i^{th} layer lattice Λ_{p_i}. It will have a form described in Lemma 3.2, given by:

$$\mathbf{G}_{\Lambda_{p_i}} = \begin{bmatrix} \mathrm{Diag}\left(\mathbf{p}_1^{(k_1)} \cdots \mathbf{p}_{i-1}^{(k_{i-1})}, \mathbf{I}_{k_i}, \mathbf{p}_{i+1}^{(k_{i+1})} \cdots \mathbf{p}_m^{(k_m)}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn-k} \end{bmatrix} \mathbf{G} \tag{3.66}$$

where $\mathbf{p}_i^{(k_i)}$ is a length-k_i vector with each element p_i. $\mathbf{G}_{\Lambda_{p_i}}$ in (3.66) gives the generator matrix for the i^{th} layer lattices, when the message input

$$\mathbf{w} = [\mathbf{w}^1, \mathbf{w}^2, \cdots, \mathbf{w}^m, \underbrace{\tilde{d}_1 \cdots \tilde{d}_m}_{mn-k}] \tag{3.67}$$

where $\mathbf{w}^i \in (\delta_i S / \langle p_i \delta_i \rangle)^{k_i}$, $\tilde{d}_i \in S^{n-k_i}$.

Since EDC lattices are constructed by some linear codes, the matrix $\mathbf{G}$ has to include the generator matrix of each linear code $\mathcal{C}^i$. Let $\tilde{\sigma}_i([\mathbf{I}_{k_i} \ \ \mathbf{B}^i_{k_i \times (n-k_i)}])$ be a generator matrix for a linear code $\mathcal{C}^i$ (without loss of generality, we consider that the linear code is systematic in this case.), then $\mathbf{G}$ is an $n \times n$ matrix defined below,

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_{k_1} & \mathbf{B}^1_{k_1 \times (n-k_1)} \\ \mathbf{I}_{k_2} & \mathbf{B}^2_{k_2 \times (n-k_2)} \\ \vdots & \vdots \\ \mathbf{I}_{k_m} & \mathbf{B}^m_{k_m \times (n-k_m)} \\ \mathbf{0} & \varpi \mathbf{I}_{n-k_1} \\ \vdots & \vdots \\ \mathbf{0} & \varpi \mathbf{I}_{n-k_m} \end{bmatrix} \tag{3.68}$$

Equation (3.68) follows from Lemma 3.5 and part of the proof of Theorem 3.1, i.e. (3.34). The generator matrix of the coarse lattice Λ' is therefore given by,

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} \text{Diag}\left(\mathbf{I}_{\sum_{j=1}^{i-1} k_j}, \mathbf{p}_i^{(k_i)}, \mathbf{I}_{\sum_{j=i+1}^{m} k_j}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn-k} \end{bmatrix} \mathbf{G}_{\Lambda_{p_i}} \tag{3.69}$$

It can be easily observed that these generator matrices are consistent with the theorems and lemmas proposed in Section 3.2.3. Note that the generator matrix for linear code $\mathcal{C}^i$ is $\tilde{\sigma}_i([\mathbf{I}_{k_i} \ \ \mathbf{B}^i_{k_i \times (n-k_i)}])$ where $\tilde{\sigma}_i$ is defined in (3.65). Theorem 3.3 establishes the theoretic fundamental for low-complexity lattice decoding, i.e. LIF, of MLNC, and states that there exists a surjective S-module homomorphism φ_i which satisfies Lemma 3.3, with kernel $\mathcal{K}(\varphi_i) = \Lambda'_i$, which plays a key role in decoding the i^{th} layer linearly combined messages. Its generator matrix has a form:

$$\mathbf{G}_{\Lambda'_i} = \begin{bmatrix} \text{Diag}(\underbrace{\mathbf{I}, \mathbf{p}_i^{(k_i)}, \mathbf{I}}_{k}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn-k} \end{bmatrix} \mathbf{G} \tag{3.70}$$

We can easily verify $\Lambda/\Lambda'_i \cong (S/\langle p_i \rangle)^{k_i}$ in terms of these generator matrices.

Scenario 2. When $\forall i = 1, 2, \cdots, m, \gamma_i \neq 1$, the primary sublattice Λ_{p_i} at each layer is constructed by a linear code over a finite chain ring $T = \delta_i S/\langle p_i^{\gamma_i} \delta_i \rangle$ [64]. A finite chain ring is a finite local principal ideal ring, and the most remarkable characteristic of a finite chain ring is that its every ideal (including $\langle 0 \rangle$) is generated by the maximal ideal, which can be linearly ordered by inclusion, and hence, forms a chain. The finite chain ring T has a unique maximal ideal, and hence the resultant residue field is $Q = \delta_i S/\langle p_i \delta_i \rangle$ with size $q = |\delta_i S/\langle p_i \delta_i \rangle|$. The chain length of the ideals is indeed the nil-potency index of p_i which is, in this case γ_i. We refer to T a (q, γ_i) chain ring.

At the i^{th} layer, the generator matrix $\mathbf{G}^i_{\text{FCR}}$ of a linear code over T has a standard form given in (3.71), where $\mathbf{I}_{k'_{i,t}}$ denotes an identity matrix with dimension $k'_{i,t}$,[1] $i = 1, 2, \cdots, m$ and $t = 0, 1, \cdots, \gamma_i - 1$. Hence, $\mathbf{G}^i_{\text{FCR}}$ has a dimension $k'_i \times n$ where $k'_i = \sum_{t=0}^{\gamma_i - 1} k'_{i,t}$. Here $Z_{t,l}$, $l = t+1, 2, \cdots, \gamma_i$, denotes a $k'_{i,t} \times k'_{i,t+1}$ ($k'_{i,\gamma_i} = n - k'_i$) matrix which is unique modulo $p_i^{\gamma_i - t}$ [65]. In (3.71), $\mathfrak{J}^*_{p_i^{\gamma_i}}$ is an upper triangular matrix with dimension $k'_i \times k'_i$, and $\mathfrak{B}_{k'_i, n-k'_i}$ has a dimension of $k'_i \times (n - k'_i)$. Note that the codeword is row spanned by $\mathbf{G}^i_{\text{FCR}}$ and all rows of $\mathbf{G}^i_{\text{FCR}}$ are linearly independent.

To study the message space of the linear codes over the finite chain ring, we first examine the kernel of the generator matrix $\mathbf{G}^i_{\text{FCR}}$. This is equivalent to finding the null space for the encoder $\mathcal{E}^i : \mathbf{w}^i \longmapsto \mathcal{C}^i$, where $\mathcal{E}^i(\mathbf{w}^i) \triangleq \mathbf{w}^i \mathbf{G}^i_{\text{FCH}}$ and $\mathbf{w}^i = [\mathbf{w}_{k'_{i,0}}, \mathbf{w}_{k'_{i,1}}, \cdots, \mathbf{w}_{k'_{i,\gamma_i - 1}}]$. Here $\mathbf{w}^i$ is grouped into blocks of size $\mathbf{w}_{k'_{i,t}}$ which corresponds to the row blocks defined

[1] Here, the index i used in $k'_{i,t}$ is the indicator of layer.

$$\mathbf{G}^i_{\mathrm{FCR}} = \left[\begin{array}{ccccc|c} \mathbf{I}_{k'_{i,0}} & Z^i_{0,1} & Z^i_{0,2} & \cdots & Z^i_{0,\gamma_{i-1}} & Z^i_{0,\gamma_i} \\ 0 & p_i\mathbf{I}_{k'_{i,1}} & p_i Z^i_{1,2} & \cdots & p_i Z^i_{1,\gamma_{i-1}} & p_i Z^i_{1,\gamma_i} \\ 0 & 0 & p_i^2\mathbf{I}_{k'_{i,2}} & \cdots & p_i^2 Z^i_{2,\gamma_{i-1}} & p_i^2 Z^i_{2,\gamma_i} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p_i^{\gamma_i-1}\mathbf{I}_{k'_{i,\gamma_i-1}} & p_i^{\gamma_i-1} Z^i_{\gamma_i-1,\gamma_i} \end{array}\right] = \left[\begin{array}{c|c} \mathfrak{J}^*_{p_i^{\gamma_i}} & \mathfrak{B}_{k'_i,n-k'_i} \end{array}\right] \tag{3.71}$$

in (3.71). In order to obtain the all-zero codeword $\mathcal{C}^i = \mathbf{0}$, we solve the homogeneous system $\mathbf{w}^i\mathbf{G}^i_{\mathrm{FCH}} = \mathbf{0}$, which gives $\mathbf{w}_{k'_{i,t}} \in p_i^{\gamma_i-t}T^{k'_{i,t}}$, $t = 0, 1, \cdots, \gamma_i - 1$. This result is based on the fact that if $d \in T^n$, then $p_i^t d = 0 \implies d \in p_i^{\gamma_i-t}T^n$. The null space of the encoder $\mathcal{E}^i$ is therefore:

$$\mathbf{w}' = [p_i^{\gamma_i}T^{k'_{i,0}}, \cdots, p_i T^{k'_{i,\gamma_i-1}}] \tag{3.72}$$

According to the first isomorphism theorem, the codeword $\mathcal{C}^i$ is isomorphic to a direct summation:

$$\begin{aligned} \mathcal{C}^i &\cong (T/p_i^{\gamma_i}T)^{k'_{i,0}} \oplus (T/p_i^{\gamma_i-1}T)^{k'_{i,1}} \oplus \cdots \oplus (T/p_iT)^{k'_{i,\gamma_i-1}} \\ &\cong (\delta_i S/\langle p_i^{\gamma_i}\delta_i\rangle)^{k'_{i,0}} \oplus (\delta_i S/\langle p_i^{\gamma_i-1}\delta_i\rangle)^{k'_{i,1}} \oplus \cdots \oplus (\delta_i S/\langle p_i\delta_i\rangle)^{k'_{i,\gamma_i-1}} \end{aligned} \tag{3.73}$$

The right-hand side of (3.73) denotes the message space $\boldsymbol{W}^i$ of the linear code over the finite chain ring T in terms of the generator matrix $\mathbf{G}^i_{\mathrm{FCR}}$. Note that each component in the direct sum of (3.73) forms another module or vector space, and the size of the t^{th} component is $q^{(\gamma_i-t)k'_{i,t}}$. This leads to the overall message size $|C| = q^{\sum_{t=0}^{\gamma_i-1}(\gamma_i-t)k'_{i,t}}$. Of course, we can obtain this result directly from the kernel of $\mathbf{G}^i_{\mathrm{FCR}}$, thus, $|C| = \prod_{t=0}^{\gamma_i-1}(p_i^tT)^{k'_{i,t}}$ which gives the same result.

Let $\tilde{\boldsymbol{p}}_i^{\gamma_i}$ be a length-k'_i vector:

$$\tilde{\boldsymbol{p}}_i^{\gamma_i} \triangleq [\mathbf{p}^{\gamma_i}_{i,(k'_{i,0})}, \mathbf{p}^{\gamma_i-1}_{i,(k'_{i,1})}, \cdots, \mathbf{p}_{i,(k'_{i,\gamma_i-1})}]$$

where $\mathbf{p}^{\gamma_i}_{i,(k'_{i,0})}$ denotes a length-$k'_{i,0}$ vector, with each component being $p_i^{\gamma_i}$. Note that $\tilde{\boldsymbol{p}}_i^{\gamma_i}$ is closely related to (3.72). Following Lemma 3.2, the generator matrix of the primary sublattice Λ_{p_i} of the i^{th} layer in this scenario has a form:

$$\mathbf{G}_{\Lambda_{p_i}} = \begin{bmatrix} \mathrm{Diag}\left(\tilde{\boldsymbol{p}}_1^{\gamma_1} \cdots \tilde{\boldsymbol{p}}_{i-1}^{\gamma_{i-1}}, \mathbf{I}_{k'_i}, \tilde{\boldsymbol{p}}_{i+1}^{\gamma_{i+1}} \cdots \tilde{\boldsymbol{p}}_m^{\gamma_m}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn-k'} \end{bmatrix} \mathbf{G} \tag{3.74}$$

where $k' = \sum_{i=1}^{m} k'_i$. The EDC lattices in this scenario are constructed by some linear codes over different finite chain rings, and the matrix $\mathbf{G}$ has to be be associated with the generator matrix of each linear code $\mathcal{C}^i$ over the finite chain ring. Let $\tilde{\sigma}_i(\mathbf{d} \cdot [\tilde{\boldsymbol{\mathfrak{I}}}^*_{p_i^{\gamma_i}} \; \tilde{\boldsymbol{\mathfrak{B}}}_{k'_i, n-k'_i}])$ be the codeword of $\mathcal{C}^i = \mathbf{w}^i \mathbf{G}^i_{\mathrm{FCR}}$ over the finite chain ring T, $\mathbf{d} \in \delta_i S^{k'_i}$. Then, $\mathbf{G}$ in (3.74) is an $mn \times n$ matrix defined below:

$$\mathbf{G} = \begin{bmatrix} \tilde{\boldsymbol{\mathfrak{I}}}^*_{p_1^{\gamma_1}} & \tilde{\boldsymbol{\mathfrak{B}}}_{k'_1, n-k'_1} \\ \tilde{\boldsymbol{\mathfrak{I}}}^*_{p_2^{\gamma_2}} & \tilde{\boldsymbol{\mathfrak{B}}}_{k'_2, n-k'_2} \\ \vdots & \vdots \\ \tilde{\boldsymbol{\mathfrak{I}}}^*_{p_m^{\gamma_m}} & \tilde{\boldsymbol{\mathfrak{B}}}_{k'_m, n-k'_m} \\ \mathbf{0} & \varpi \mathbf{I}_{n-k'_1} \\ \vdots & \vdots \\ \mathbf{0} & \varpi \mathbf{I}_{n-k'_m} \end{bmatrix} \tag{3.75}$$

Hence, we are able to construct Λ_{p_i}, and hence the EDC lattice Λ for this scenario based on the generator matrices presented above. Note that message space of each layer follows from (3.73), and $k'_{i,t}$ should be selected such that

$$\gamma_i k_i = \sum_{t=0}^{\gamma_i - 1} (\gamma_i - t) k'_{i,t} \tag{3.76}$$

in order to guarantee the consistency to the message size of the i^{th} layer EDC lattices defined in (3.63). It is easy to prove that there exists $k'_{i,t} \in \mathbb{Z}^+$, $\forall t = 0, 1, \cdots, \gamma_i - 1$, satisfying (3.76).

The generator matrix of the coarse lattice Λ' is given by,

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} \mathrm{Diag}\left(\mathbf{I}_{\sum_{j=1}^{i-1} k'_j}, \tilde{\boldsymbol{p}}_i^{\gamma_i}, \mathbf{I}_{\sum_{j=i+1}^{m} k'_j} \right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn-k} \end{bmatrix} \mathbf{G}_{\Lambda_{p_i}} \tag{3.77}$$

Following (3.77), it is obvious that $\Lambda/\Lambda' \cong \boldsymbol{W}^1 \oplus \cdots \oplus \boldsymbol{W}^m$. The generator matrix for Λ'_i has a form:

$$\mathbf{G}_{\Lambda'_i} = \begin{bmatrix} \mathrm{Diag}\left(\mathbf{I}_{\sum_{j=1}^{i-1} k'_j}, \tilde{\boldsymbol{p}}_i^{\gamma_i}, \mathbf{I}_{\sum_{j=i+1}^{m} k'_j} \right) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{mn-k} \end{bmatrix} \mathbf{G} \tag{3.78}$$

which will be used for LIF detection.

Every ideal of T is generated by the maximal ideal, which forms a chain with chain length γ_i. Hence, the residue field Q plays an important role in producing the linear codes over T. We now consider a matrix in the form of:

$$\mathbf{G}_{\mathrm{D}}^{i} = \mathrm{Diag}\left(\mathbf{p}_{i,(k'_{i,0})}^{0}, \cdots, \mathbf{p}_{i,(k'_{i,\gamma_i-1})}^{\gamma_i-1}\right) \begin{bmatrix} \mathbf{g}_{k'_{i,0}}^{i} \\ \mathbf{g}_{k'_{i,1}}^{i} \\ \vdots \\ \mathbf{g}_{k'_{i,\gamma_i-1}}^{i} \end{bmatrix} \tag{3.79}$$

where $\mathbf{g}_{k'_{i,t}}^{i} \in Q^{*}_{k'_{i,t}\times n}$, and $Q^{*}_{k'_{i,t}\times n}$ is a $k'_{i,t} \times n$ matrix with each entry over the coset representative of the residue field $Q = \delta_i S/\langle p_i\delta_i\rangle$. Each row of $\mathbf{G}_{\mathrm{D}}^{i}$ has to satisfy the condition that none of its rows are linear combinations of the other rows. The message space of $\mathbf{G}_{\mathrm{D}}^{i}$ could be partitioned into $\gamma_i - 1$ levels. We first define the vector $\boldsymbol{\beta}_{k'_{i,t}}^{(j)} = [\beta_1^{(j)}, \beta_2^{(j)}, \cdots, \beta_{k'_{i,t}}^{(j)}]$, when $t = 0$, where $j = 0, 1, \cdots \gamma_i - 1$, is the level indicator, and $\boldsymbol{\beta}_{k'_{i,t}}^{(j)} = [\beta_{k'_{i,t-1}+1}^{(j)}, \beta_{k'_{i,t-1}+2}^{(j)}, \cdots, \beta_{k'_{i,t}}^{(j)}]$ when $t = 1, 2, \cdots, \gamma_i - 1$. Accurately $\boldsymbol{\beta}_{k'_{i,t}}^{(j)}$ represents a length-$k'_{i,t}$ segment of the j^{th} level message over the vector space $Q^{k'_{i,t}}$. The full message space of the j^{th} level is given by,

$$\boldsymbol{\beta}^{(j)} = [p_i^{j}\boldsymbol{\beta}_{k'_{i,0}}^{(j)}, p_i^{j-1}\boldsymbol{\beta}_{k'_{i,1}}^{(j)}, p_i^{j-2}\boldsymbol{\beta}_{k'_{i,2}}^{(j)}, \underbrace{\mathbf{0}\cdots\mathbf{0}}_{k'_i-\sum_{t=0}^{j} k'_{i,t}}] \tag{3.80}$$

where the powers of p_i cannot be negative integers. Hence, the message space of $\mathbf{G}_{\mathrm{D}}^{i}$ is $W^i = \boldsymbol{\beta}^{(0)} + \boldsymbol{\beta}^{(1)} + \cdots + \boldsymbol{\beta}^{(\gamma_i-1)}$. The codewords $\mathcal{C}^i$ can be produced by

$$\begin{aligned} \mathcal{C}^i = W^i\mathbf{G}_{\mathrm{D}}^{i} &= \left(\boldsymbol{\beta}^{(0)} + \boldsymbol{\beta}^{(1)} + \cdots + \boldsymbol{\beta}^{(\gamma_i-1)}\right)\mathbf{G}_{\mathrm{D}}^{i} \\ &= \mathbf{c}_0^i + \mathbf{c}_1^i p_i + \cdots + \mathbf{c}_{\gamma_i-1}^i p_i^{\gamma_i-1} \end{aligned} \tag{3.81}$$

Since none of the rows of $\mathbf{G}_{\mathrm{D}}^{i}$ are linear combinations of the other rows, $\mathbf{c}_t^i$ is therefore row spanned by

$$\mathbf{g}_{\mathbf{c}_t^i} = \left[\mathbf{g}_{k'_{i,0}}^{i};\ \ \mathbf{g}_{k'_{i,1}}^{i};\ \ \cdots;\ \ \mathbf{g}_{k'_{i,t}}^{i}\right] \tag{3.82}$$

It is obvious that $\mathbf{c}_t^i, t = 0, 1, \cdots, \gamma_i - 1$ forms a set of nested codes $\mathbf{c}_0^i \subseteq \mathbf{c}_1^i \subseteq \cdots \subseteq \mathbf{c}_{\gamma_i-1}^i$ over Q^*. Following the Q-adic decomposition theorem of finite chain rings [65] [64], we assert that the codeword $\mathcal{C}^i$ in (3.81) generated by $\mathbf{G}_{\mathrm{D}}^{i}$ is indeed over T.

In terms of (3.79) and (3.80), the message space corresponding to $\mathbf{g}_{k'_{i,t}}^{i}$ should be written as:

$$W_t^i = \sum_{j=t}^{\gamma_i-1} p_i^{j-t}\boldsymbol{\beta}_{k'_{i,t}}^{(j)} \tag{3.83}$$

this complies with the Q-adic decomposition and leads to the result that the message space corresponding to $\mathbf{g}_{k'_{i,t}}^{i}$ is $(T/\langle p_i^{\gamma_i-t}\rangle)^{k'_{i,t}}$. This implies that the right-hand side of

(3.73) is precisely the message space of $\mathbf{G}_{\mathrm{D}}^{i}$. Mathematically the primary sublattices Λ_{p_i} can also be represented in the form below:

$$\Lambda_{p_i} = \bigcup \left\{ \underbrace{\sum_{j=0}^{\gamma_i - 1} \sum_{\ell=1}^{\mathscr{K}_j^i} p_i^j \beta_\ell^{(j)} \mathbf{g}_\ell^i}_{(52)} + p_i^{\gamma_i} S^n | \mathbf{g}_\ell^i \in Q_{1\times n} \right\} \tag{3.84}$$

where $\mathscr{K}_j^i = k'_{i,0} + \cdots + k'_{i,j}$. It is interesting to see that (3.84) has the same structure as complex construction D. Now we conclude that the primary S-sublattices constructed by a linear code over a finite chain ring subsumes construction D.

Based on this result, we may now construct EDC lattices for this scenario using a set of nested linear codes over a finite field. Let $\mathbf{g}_{(n-k'_i)}^{i} \in Q^{*}_{n-k'_i \times n}$ be an $(n - k'_i) \times n$ matrix, then the $\mathbf{G}$ matrix is:

$$\mathbf{G} = \left[{\mathbf{G}_{\mathrm{D}}^{1}}^{T}, \quad \cdots, \quad \mathbf{G}_{\mathrm{D}}^{mT}, \quad \varpi {\mathbf{g}_{(n-k'_i)}^{1}}^{T}, \quad \cdots, \quad \varpi {\mathbf{g}_{(n-k'_i)}^{m}}^{T} \right]^{T} \tag{3.85}$$

Scenario 3. This corresponds to a hybrid case of Scenario 1 and 2, and we give the following summaries:

1. $m = 1$, $\gamma_1 = 1$, then the EDC lattice in (3.62) is a complex construction A lattice which is indecomposable.

2. $m = 1$, $\gamma_1 > 1$, $\gamma_1 \in \mathbb{Z}^+$ then the EDC lattice in (3.62) is a complex construction D lattice which is indecomposable.

3. $m > 1, m, \gamma_i \in \mathbb{Z}^+, i = 1, 2, \cdots, m$, then the EDC lattice in (3.62) is decomposable, and consists of some sublattices constructed by either construction A or D.

Note that in 3), a new class of lattices over S is generated by a number of linear codes over either finite field or chain ring, which generalises the scenario 1 and 2. Scenario 3 suggests that the design of EDC lattices is very flexible, and we also give more detailed discussion about why EDC lattices are good at low-complexity decoding and throughput improvement for WPLNC in the next sections.

Nominal coding gain and Kissing number

In this section, we study the nominal coding gain and kissing number of the EDC lattices for all three scenarios. The definition such as the minimum-norm coset leaders and minimum Euclidean weight of the codeword follows from [49].

Scenario 1. We first study the nominal coding gain and kissing number of the i^{th} layer primary sublattices in this scenario. Following (3.64) and (3.65), we know that $\mathcal{C}^i$ is a linear code of length n over $\delta_i S/p_i\delta_i S$. Thus, $\mathbf{c}^i = (c_1^i + \langle\varpi\rangle, \cdots, c_n^i + \langle\varpi\rangle) \in \mathcal{C}^i$. We denote $\omega^{(i)}(\mathbf{c}^i)$ the Euclidean weight of a codeword $\mathbf{c}^i$ in $\mathcal{C}^i$, and $\omega^{(i)}_{\min}(\mathcal{C}^i)$ the minimum Euclidean weight of non-zero codewords in $\mathcal{C}^i$. Let ϑ be a scaling factor depending on which PID is used, and $N(\omega^{(i)}_{\min}(\mathcal{C}^i))$ be the number of codewords in $\mathcal{C}^i$ with the minimum Euclidean weight $\omega^{(i)}_{\min}(\mathcal{C}^i)$.

Proposition 3.1. *Let $\mathcal{C}^i$ be a linear code over $\delta_i S/p_i\delta_i S$, and Λ_{p_i}/Λ' the primary quotient lattice system of the i^{th} layer constructed by $\mathcal{C}^i$, $\Lambda_{p_i} \supseteq \Lambda'$, then the nominal coding gain is given by:*

$$\varrho(\Lambda_{p_i}/\Lambda') = \frac{\omega^{(i)}_{\min}(\mathcal{C}^i)}{\vartheta |p_i|^{2(1-\frac{k_i}{n})}|\delta_i|^2} \tag{3.86}$$

and the kissing number is:

$$K(\Lambda_{p_i}/\Lambda') = \begin{cases} N(\omega^{(i)}_{\min}(\mathcal{C}^i)) \left(\frac{\mathcal{N}_{\mathcal{U}(S)}}{|p_i|^2-1}\right)^{\frac{\omega^{(i)}_{\min}(\mathcal{C}^i)}{|\delta_i|^2}}, & |p_i|^2 - 1 \leq \mathcal{N}_{\mathcal{U}(S)} \\ N(\omega^{(i)}_{\min}(\mathcal{C}^i)), & \text{Otherwise} \end{cases} \tag{3.87}$$

Proof: See Section 3.2.8. ■

Here $\mathcal{N}_{\mathcal{U}(S)}$ represents the number of units in S.

It is of interest to study the nominal coding gain and kissing number of Λ/Λ' in terms of the m linear codes $\mathcal{C}^i$. Following the proof of Theorem 3.1, and the descriptions in Section 3.2.4, $\tilde{\mathbf{c}} = \mathbf{c}^1 + \mathbf{c}^2 + \cdots + \mathbf{c}^m$, $\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}$ and $\tilde{\mathcal{C}} \in (S/\langle\varpi\rangle)^n$. Thus, the nominal coding gain of EDC lattices is determined by the m linear codes $\mathcal{C}^i$ over $\delta_i S/p_i\delta_i S$, $i = 1, 2, \cdots, m$.

Proposition 3.2. *Let $\mathcal{C}^1, \cdots, \mathcal{C}^m$ be m linear codes over $\delta_i S/p_i\delta_i S$, $i = 1, 2, \cdots, m$, respectively. Let $\tilde{\mathbf{c}} = \mathbf{c}^1 + \mathbf{c}^2 + \cdots + \mathbf{c}^m$, $\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}$ and $\mathbf{c}^i \in \mathcal{C}^i$. The nominal coding gain of the EDC lattices Λ/Λ' in scenario 1 is given by*

$$\varrho(\Lambda/\Lambda') = \frac{\omega_{\min}(\tilde{\mathcal{C}}) \prod_{\ell=2}^{m} |p_j|^{\frac{2(k_\ell - k_1)}{n}}}{\vartheta |p_1|^{2(1-\frac{k_1}{n})}|\delta_1|^2} \tag{3.88}$$

where $k_1 \leq k_2 \leq \cdots \leq k_m$.

Proof: See Section 3.2.8. ■

Scenario 2. This corresponds to the case where $\gamma_i > 1$, $\gamma_i \in \mathbb{Z}$ for $i = 1, 2, \cdots, m$. The primary sublattice of the i^{th} layer can be constructed by a linear code $\mathcal{C}^i$ over a finite chain ring $\delta_i S/\langle \varpi \rangle$, where $\delta_i = \frac{\varpi}{p_i^{\gamma_i}}$. This follows immediately from (3.64) and (3.65). Here, we are more concerned with the nominal coding gain and kissing number when the i^{th} primary sublattice is constructed by a set of nested linear codes over the residue field Q, since the linear code over a finite field is easier to generate. Let $\mathcal{C}^{i,0} \subseteq \cdots \subseteq \mathcal{C}^{i,\gamma_i - 1}$ be nested linear codes of length-n over Q, where $\mathcal{C}^{i,t}$ is an $[n, \sum_{\ell=0}^{t} k'_{i,\ell}]$ linear code for the t^{th} nested code at the i^{th} layer, and we denote $\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})$ the minimum Euclidean weight of non-zero codewords in $\mathcal{C}^{i,t}$. We have:

Proposition 3.3. *Let $\mathcal{C}^{i,0} \subseteq \cdots \subseteq \mathcal{C}^{i,\gamma_i-1}$ be γ_i nested linear codes of length-n over Q, and Λ_{p_i}/Λ' be the primary quotient lattice of the i^{th} layer constructed from $\mathcal{C}^{i,t}$, $t = 0, 1, \cdots, \gamma_i - 1$, then the nominal coding gain of the i^{th} layer is lower bounded by*

$$\varrho(\Lambda_{p_i}/\Lambda') \geq \frac{|p_i|^{\frac{2}{n}\sum_{t=0}^{\gamma_i-1}(\gamma_i - t)k'_{i,t}} \min_{0 \leq t \leq \gamma_i - 1}\{|p_i|^{2t}\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})\}}{\vartheta|\varpi|^2} \tag{3.89}$$

and the kissing number is upper bounded by:

$$K(\Lambda_{p_i}/\Lambda') \leq \begin{cases} \sum_{t=0}^{\gamma_i-1} N_t(\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})) \left(\frac{\mathcal{N}_{\mathcal{U}(S)}}{|p_i|^2 - 1}\right)^{\frac{\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})}{|\delta_i|^2}}, & |p_i|^2 - 1 \leq \mathcal{N}_{\mathcal{U}(S)} \\ \sum_{t=0}^{\gamma_i-1} N_t(\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})), & \text{Otherwise} \end{cases} \tag{3.90}$$

Proof: See Section 3.2.8. ■

It is of interest to study the nominal coding gain of Λ/Λ' in this scenario. If each primary sublattice is constructed via a set of nested linear codes over a finite field $Q = \delta_i S/\langle p_i \delta_i \rangle$ for the i^{th} layer, the nominal coding gain $\varrho(\Lambda/\Lambda')$ will be related to overall $\sum_{i=1}^{m} \gamma_i$ linear codes since there are γ_i nested linear codes for each i. Let $\tilde{\mathcal{C}}$ be a composite code such that $\tilde{\mathbf{c}} = \mathbf{c}^1 + \cdots + \mathbf{c}^m$ where $\mathbf{c}^i = \mathbf{c}^{i,0} + p_i \mathbf{c}^{i,1} + \cdots + p_i^{\gamma_i - 1}\mathbf{c}^{i,\gamma_i-1}$. Hence, $\mathcal{C}^i \in \delta_i S/\langle \varpi \rangle$ and $\tilde{\mathcal{C}} \in S/\langle \varpi \rangle$. We denote $\omega_{\min}(\tilde{\mathcal{C}})$ the minimum Euclidean weight of non-zero codewords in $\tilde{\mathcal{C}}$, then:

Proposition 3.4. *Let $\mathcal{C}^{i,0} \subseteq \cdots \subseteq \mathcal{C}^{i,\gamma_i-1}$ be γ_i nested linear codes of length-n over Q, and let $\tilde{\mathcal{C}}$ be a composite code such that $\tilde{\mathbf{c}} = \mathbf{c}^1 + \cdots + \mathbf{c}^m$ where $\mathbf{c}^i = \mathbf{c}^{i,0} + p_i\mathbf{c}^{i,1} + \cdots + p_i^{\gamma_i-1}\mathbf{c}^{i,\gamma_i-1}$. The nominal coding gain for Λ/Λ' in scenario 2 is given by:*

$$\begin{aligned} \varrho(\Lambda/\Lambda') &= \frac{\omega_{\min}(\tilde{\mathcal{C}})}{(V(\mathcal{V}(\Lambda)))^{\frac{1}{n}}} \\ &= \frac{\omega_{\min}(\tilde{\mathcal{C}}) \prod_{i=1}^{m} |p_i|^{2\sum_{t=0}^{\gamma_i-1}(\gamma_i - t)\frac{k'_{i,t}}{n}}}{\vartheta|\varpi|^2} \end{aligned} \tag{3.91}$$

Proof: See Section 3.2.8. ■

Scenario 3. As explained in the preceding section, in this case, $\gamma_i \geq 1$, $\gamma_i \in \mathbb{Z}$, and hence the EDC lattice consists of a number of primary sublattices which can be constructed by linear codes over either finite field or finite chain ring. The nominal coding gain and kissing number of the primary sublattices in each case have been derived in Proposition 3.1 and Proposition 3.3. We are more interested in the nominal coding gain of Λ/Λ' in this scenario. Again, we consider the primary sublattices of scenario 2 is constructed over a set of nested linear codes. Let $\tilde{\mathcal{C}}$ be a composite code such that $\tilde{\mathbf{c}} = \mathbf{c}^1 + \cdots + \mathbf{c}^m$ where

$$\mathbf{c}^i = \begin{cases} \mathbf{c}^i, \;\; \mathcal{C}^i \in \delta_i S/p_i \delta_i S, & \gamma_i = 1 \\ \mathbf{c}^{i,0} + p_i \mathbf{c}^{i,1} + \cdots + p_i^{\gamma_i - 1} \mathbf{c}^{i,\gamma_i - 1}; \;\; \mathcal{C}^{i,t} \in Q, & \gamma_i > 1 \end{cases}$$

We can easily prove that $\varrho(\Lambda/\Lambda')$ has similar form as (3.91) if we set $k'_{i,0} = k_i$ for $\gamma_i = 1$.

3.2.5 Iterative detection of EDC and the EXIT chart analysis

In this section we present an iteration-aided multistage decoding approach specifically designed for EDC, which provides a feasible way of improving the performance of decoding the linear combinations, and also of increasing the overall rate with low decoding-complexity. In the remainder of the paper, we consider S to be a ring of Eisenstein integers $\mathbb{Z}[\omega]$. However, the results can be readily extended to other PIDs.

Section 3.2.4 clearly reveals the possible encoding structure for EDC. Recalling the definition for EDC, we know that the map $\tilde{\sigma} : S^n \longmapsto (S/\langle p_1^{\gamma_1} \rangle)^n \oplus (S/\langle p_2^{\gamma_2} \rangle)^n \oplus \cdots \oplus (S/\langle p_m^{\gamma_m} \rangle)^n$ is a natural projection of a surjective ring homomorphism $\sigma : S \longmapsto S/\langle p_1^{\gamma_1} \rangle \times S/\langle p_2^{\gamma_2} \rangle \times \cdots \times S/\langle p_m^{\gamma_m} \rangle \longleftrightarrow \mathbb{F}_{\tilde{p}_1} \times \cdots \times \mathbb{F}_{\tilde{p}_m}$ by applying it element-wise [66] ($\gamma_i = 1$, $\forall i = 1, 2, \cdots, m$). Note that in this case, σ is actually an *f.g.* Abelian group homomorphism. It is easy to see that each level $S/\langle p_i \rangle$ is coded by an $[n, k_i]$ linear code $\mathcal{C}^i$ over $\mathbb{F}_{\tilde{p}_i}$ (a finite field or finite chain ring determined by $\tilde{p}_i$).

The Type-1 Eisenstein primes are those primes $p \in \mathbb{Z}$ which either have a form $6j + 5$, $j \in \mathbb{Z}$, or $p = 2$. Their associates are also categorised as Type-1. The Type-2 Eisenstein primes have the form $\tau = a + b\omega$, $a, b \neq 0$ where the norm $\mathcal{N}(\tau)$ of τ is a prime $p \in \mathbb{Z}$ satisfying $p \equiv 1 \bmod 6$. Note that if $\tau = a + b\omega$ is a prime in $\mathbb{Z}[\omega]$, $\tau' = b + a\omega$ is also a prime in $\mathbb{Z}[\omega]$. Hence, τ and τ' are distinct primes categorised as Type-2. Together with the Type-3 Eisenstein primes, $\varpi \in \mathbb{Z}[\omega]$ can be uniquely decomposed into:

$$\varpi = \mathcal{U}(\mathbb{Z}[\omega]) 2^{\varrho} \prod_{i=1}^{\kappa_1} \tau_i^{\mu_i} \prod_{j=1}^{\kappa_2} \tau_j'^{\eta_j} \prod_{k=1}^{\kappa_3} p_k^{\beta_k} \cdot (1 + 2\omega)^{\varsigma} \tag{3.92}$$

Accordingly, $\mathbb{Z}[\omega]/\langle 2 \rangle \cong \mathbb{F}_{2^2}$, $\mathbb{Z}[\omega]/\langle \tau \rangle \cong \mathbb{F}_{\mathcal{N}(\tau)}$, $\mathbb{Z}[\omega]/\langle p \rangle \cong \mathbb{F}_{p^2}$, $\mathbb{Z}[\omega]/\langle 1 + 2\omega \rangle \cong \mathbb{F}_3$.

Soft detector for EDC

Section 3.2.3 gives a general decoding method LIF for MLNC, based on the optimised scaling factor α, S-integer coefficient vectors $\tilde{\mathbf{a}}_i$, and a good EDC lattice quantiser, e.g. a Viterbi decoder with modified metrics (see Section 3.2.8). Thus, when EDC is employed in MLNC, LIF is also feasible. In this section, we explore another detection approach designed specifically for the EDC-based MLNC (which follows from the structure of the EDC lattices). Especially an iterative detector is developed, which exploits the multilevel structure gain of EDC by using multistage decoding.

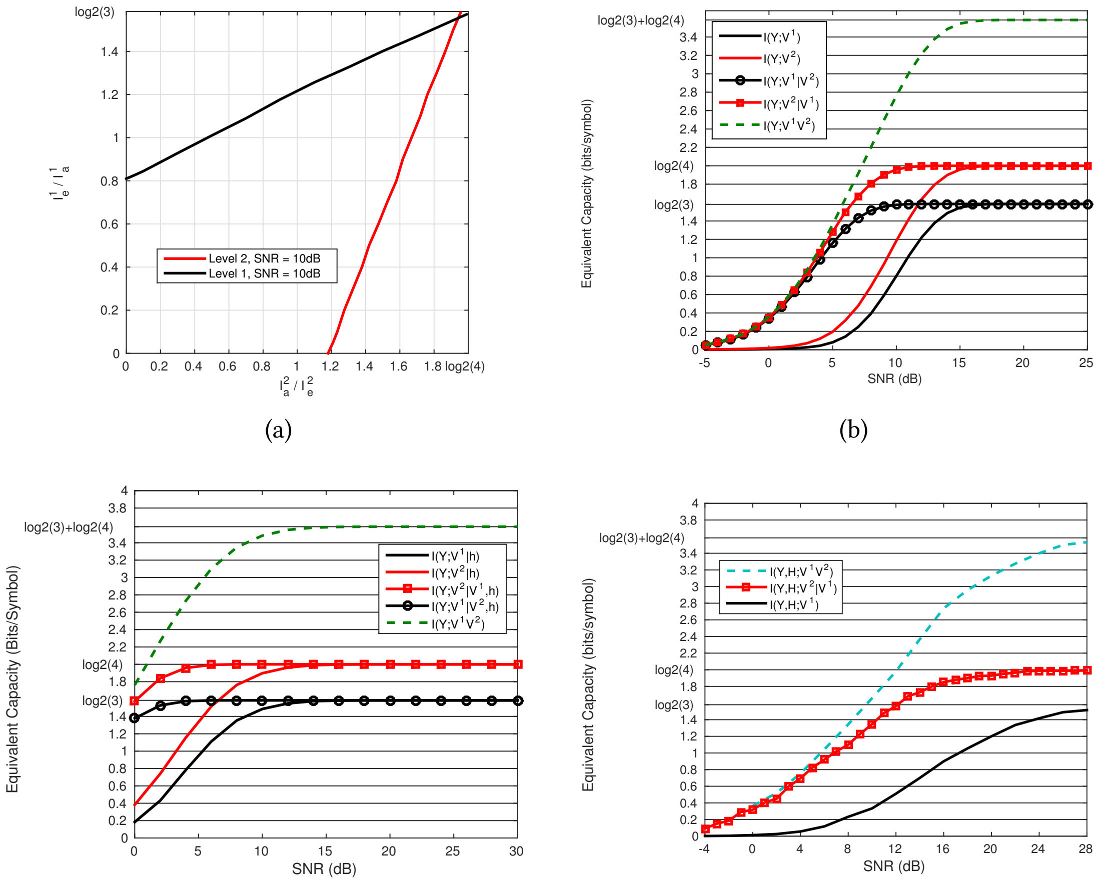
First, we consider the non-iterative multistage decoding. The detector tries to decode the linear function of each level stage-by-stage, with the aid of the *a priori* information from the preceding layers. The detection structure is similar to the point-to-point multi-level codes, e.g. [67,68] whereas here the *a priori* information is the soft estimation. We develop a layered soft detector (LSD) which calculates the a posteriori L-vector (a vector of Log-likelihood ratio) for each layer with the aid of the multiple *a priori* L-vectors. The detailed derivation is given in Section 3.2.8.

The LSD decodes the linear function of each layer over the corresponding non-binary finite field, and hence the *a priori* information of each layer is no longer a scalar value. We define the *a priori* information $\mathbf{A}^i$ to be a vector-based random variable with realisation:

$$\mathbf{a}^i = \left[\log\left(\frac{\Pr(\xi|V^i = v_1^i)}{\Pr(\xi|V^i = 0)}\right)\cdots\log\left(\frac{\Pr(\xi|V^i = v_{\tilde{p}_i-1}^i)}{\Pr(\xi|V^i = 0)}\right)\right] \tag{3.93}$$

where V^i denotes the possible linear combinations at the i^{th} level, which is a uniformly distributed random variable whose k^{th} realisation is $v_k^i \in \mathbb{F}_{\tilde{p}_i}$, $k = 1, 2\cdots\tilde{p}_i - 1$. $\Pr(\xi|V^i = v_k^i)$ is the probability of the *a priori* channel outputs $\Xi = \xi$ given the event $V^i = v_k^i$. Assume that $w_j^i \in \mathbb{F}_{\tilde{p}_i}$, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, L$ to be the message of the i^{th} level and the j^{th} source, the linear function is defined by $f^i(w_1^i, \cdots, w_L^i) = \bigoplus_{\ell=1}^{L} a_\ell^i w_\ell^i$ over $\mathbb{F}_{\tilde{p}_i}$. Note that the integer coefficient a_ℓ^i can be determined either by the lattice reduction approach as introduced in [49,69] over the i^{th} quotient lattice Λ/Λ_i' as defined in Theorem 3.3, or by the maximum mutual information criterion as described later.

In the multistage iterative decoding, the proposed LSD outputs the extrinsic L-vector $\mathbf{e}^i$ for the i^{th} level, based on the *a priori* L-vector $\mathbf{a}^j, j \in \{1, \cdots, m\}, j \neq i$. Assume that there is a two level EDC and the decoding proceeds from layer 1 (which is regarded as the 1^{st} stage decoding) to layer 2 (the 2^{nd} stage decoding). The extrinsic outputs of layer 1 feed into layer 2 to assist the 2^{nd} stage decoding. With the aid of the *a priori* L-value, layer 2 estimates and forwards the extrinsic information (which serves as the *a priori* information of layer 1) to layer 1. The process is repeated and all layers are activated in turn for the second and subsequent iterations. We refer to this approach as the iterative MSD (IMSD) scheme for MLNC. The detection process is similar to iterative decoding of multilevel codes, e.g. [70] whereas the nature of the detection is different. As



(a)

(b)

(c)

(d)

Figure 3.12: (a) The EXIT function for each layer. (b) Achievable information rates of the linear combinations at each layer; $h_1 = h_2 = 1$. (c) Achievable rates of the linear combinations at each layer with fixed fading coefficients $h_1 = -1.17 + 2.15 * 1i$ and $h_2 = 1.25 - 1.63 * 1i$. (d) Achievable rates of the linear combinations at each layer with Rayleigh fading.

the iteration proceeds, each layer will produce more reliable extrinsic L-vector $\mathbf{e}^i$ which also serves as the *a priori* information of the soft-in soft-out non-binary decoder for the corresponding $\mathcal{C}^i$.

Non-binary EXIT chart analysis

We now evaluate the extrinsic information transfer characteristics of the soft detector developed in Section 3.2.5, based on the non-binary case. Alexei *et al.* [71] has proved, based on the binary iterative system, that the extrinsic information E_k (the k^{th} time instant) of an *a posteriori* probability (APP) decoder contains the same amount of in-

$$I_{\mathrm{E}} = H(V) + \mathbb{E}\Big[\frac{1}{N}\sum_{k=1}^{N}\sum_{\forall v}\Pr(V_k = v|\mathbf{y}, \mathbf{z}_{\backslash k}) \cdot \log(\Pr(V_k = v|\mathbf{y}, \mathbf{z}_{\backslash k}))\Big] \tag{3.94}$$

formation as the physical channel outputs $\mathbf{Y}$ and the outputs of the *a priori* channel $\mathbf{Z}_{\backslash k}$. We can prove that when the extrinsic outputs are non-binary-based, this theorem also holds. In this case, E_k becomes the vector-based random variable $\mathbf{E}_k$, and can be expressed as:

$$I(V_k|\mathbf{E}_k) = I(V_k|\mathbf{Y}, \mathbf{Z}_{\backslash k})$$

The proof [72] is based on the fact that $\Pr(V_k = v|\mathbf{e}_k) = \Pr(V_k = v|\mathbf{y}, \mathbf{z}_{\backslash k})$. The average extrinsic information $I_{\mathrm{E}} = \frac{1}{N}\sum_{k=1}^{N} I(V_k; \mathbf{E}_k)$ can be obtained by:

Figure 3.12(a) illustrates the extrinsic transfer characteristics for a two-level EDC lattice over the 2-WRC, where $\varpi = 2 + 4\omega = 2(1 + 2\omega)$. Based on the definition of EDC and (3.92), the linear codes $\mathcal{C}^1$ and $\mathcal{C}^2$ are over $\mathbb{F}_3$ and a binary extension field $\mathbb{F}_{2^2}$, respectively. The extrinsic information I_e^1 for the linear combinations of the 1st level depends only on the *a priori* information I_a^2 from the 2nd level, and similarly for I_e^2. It can be observed in Figure 3.12(a) that there is an increase of the average extrinsic information I_e^1 around 0.8 at 10dB when the soft detector has the ideal *a priori* information at the 2nd level, compared to the non-iterative case. Hence, the iteration-aided multistage detection implies a large potential to improve the reliability of decoding the linear combinations at each level. Due to space limitation for this paper, we show here results only for $h_1 = h_2 = 1$. However the results can be easily extended to the faded MAC. Note that the optimal linear functions $f^1, \cdots, f^m$ should be selected in terms of:

$$f^1 \cdots f^m = \underset{f^1 \dots f^m}{\arg\max}\, I(Y; V^1V^2 \cdots V^m) \tag{3.95}$$

which maximises the achievable rate. Note that V^i is a random variable with its outcomes from the linear function f^i. Hence, the conditional probability density $\Pr(Y|V^i)$ is a function of the messages $w_j^i, j = 1, 2, \cdots, L$. Figure 3.12(b) gives the numerical integration results for the achievable rates at each level. It can be observed that the mutual information chain rule is satisfied, which gives theoretical support for multistage iterative decoding. Figure 3.12(b) also well matches the EXIT chart results in Figure 3.12(a), e.g. the extrinsic information of the linear combinations for the first level is around $I(Y; V^1) = 0.8$ and $I(Y; V^1|V^2) = 1$ at 10dB which precisely match the black line in Figure 3.12(a). Figure 3.12(c) and 3.12(d) give the achievable information rates of the linear combinations at each level based on the fixed fading and Rayleigh fading, respectively. The detailed calculation of the these are described in Section 3.2.8.

It is seen that the maximum achievable rates for the network coded linear combinations are $\mathcal{R}^{(1)} = \log_2 3$ and $\mathcal{R}^{(2)} = \log_2 4$ for level 1 and 2. The allowable rate at a certain level is higher when the *a priori* information from another layer is available. We assume two memory 3, 1/2-rate convolutional codes are used at both levels (over $\mathbb{F}_3$ and $\mathbb{F}_{2^2}$ respectively). EDC lattices achieve overall rate $\frac{1}{2}\log_2(12)$, with the number of trellis states 27 and 64 at the corresponding levels. However, a single convolutional code over ring R_{12} needs 1728 trellis states. The complexity reduction is obvious.

3.2.6 Simulation results

In this section, we evaluate the performance of the MLNC scheme, based on the detection approaches proposed. These results give strong support for the theorems and lemmas developed in previous sections. In this paper, we focus mainly on the applications of EDC lattices in MLNC. However, it is not necessarily limited to EDC lattices since MLNC design applies in principle to the general case. For example, high coding gain lattice codes (e.g. complex low density lattice codes [59, 73] and signal codes [74]) which are directly designed in the geometric space can be used in the MLNC framework. This is interesting and will be investigated in our later work.

We are mainly concerned (in this paper) with the performance of the multiple access channel (MAC) of the 2-WRC, which can be viewed as the building block for more complicated network topologies. All simulations are based on a two-layer EDC lattice which has the same configuration. Thus, the two layers are constructed via linear codes $\mathcal{C}^1 \in \mathbb{F}_3$ and $\mathcal{C}^2 \in \mathbb{F}_{2^2}$. The linear codes at both layers are non-binary convolutional codes, with their generator polynomials defined in Table 3.1. Note that the decoder of the non-binary convolutional codes is based on the maximum a posterior (MAP) probability criteria and modified BCJR algorithm, where the soft output of the component symbols is produced. We do not give detailed explanation of the decoding in this paper since it is not our main concern, but we will provide the algorithm when requested. Unless otherwise stated, the convolutional decoder employs the same algorithm in the sequel.

Table 3.1: Code type and code rate assigned for each level.

i	$\mathbf{g}(D)$
1	$[-2\omega^2 + 2\omega^2 D^3, 2\omega^2 + (-2\omega^2)D + 2\omega^2 D^3]$
2	$\begin{bmatrix} -2 + (1-\omega)D^2 + (-2)D^3 \\ -2 + (-2)D + (-2)D^2 + (1-\omega^2)D^3 \end{bmatrix}$

Figure 3.13 depicts the symbol-error rates and frame-error rates for EDC-based MLNC as a function of SNR, where the soft detection approach is used. We examine the performance with and without multistage decoding when iterations and fading are not considered. The convolutional code at the i^{th} level is defined as a $[2(\iota^i + \nu^i), \iota^i]$ linear block code, where ι^i and ν^i denote the data and memory length, respectively. Therefore, the

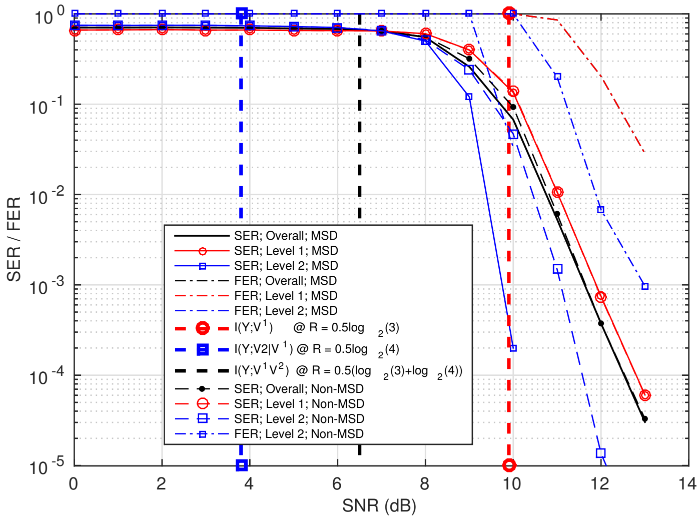


Figure 3.13: SER and FER performance for an MLNC constructed from a two layer EDC lattice; Soft detection; Multistage decoding/Non-multistage decoding; $\mathcal{R}^{(1)}_{\text{mes}} = \frac{1}{2}\log_2 3$; $\mathcal{R}^{(2)}_{\text{mes}} = \frac{1}{2}\log_2 4$; $h_1 = h_2 = 1$.

overall message rate is given by:

$$\mathcal{R}_{\text{mes}} \approx \frac{1}{2}(\log_2 3 + \log_2 4) \text{ bits/symbol}$$

Note that we use the approximation sign here since the actual coding rate is smaller than $\frac{1}{2}$ due to the tail effect of memory. When the block length is sufficient large, this effect can be ignored. Without multistage decoding, it is observed from Figure 3.13 that the SER gap between layer 1 and 2 is around 0.8 dB at BER=10^{-4}, and layer 1 is 8 dB from the capacity of layer 1. When multistage decoding is performed from layer 1 to layer 2, we expect that the SER performance of layer 2 can be improved as a result of the additional *a priori* soft information from layer 1. Note that layer 2 operates over $\mathbb{F}_{2^2}$ whereas its *a priori* soft information is over $\mathbb{F}_3$. The simulation results confirm this anticipation in that the SER of layer 2 has 2 dB gain over non-MSD at 10^{-5}. However this leads to only slightly better overall performance. When multistage processing starts from layer 1, it is obvious that MSD and non-MSD should give approximately the same performance at layer 1. The overall performance is dominated by the layer which has the worst SER performance over all layers, and in this case, it is layer 1. This explains the reason why the performance improvement of layer 2 gives small contribution in the overall SER.

To further increase network throughputs, and examine the performance of MSD based on the asymmetric coding rates over each level, the rate of layer 2 is set to $\mathcal{R}^{(2)} = \frac{3}{4}$. Thus, the sublattice Λ_{p_2} is constructed via a higher rate linear code. The overall message rate is given by

$$\mathcal{R}_{\text{mes}} \approx \frac{1}{2}\log_2 3 + \frac{3}{4}\log_2 4 \text{ bits/symbol}$$

Note that the SER curve of level 1 (red dashed circle) without MSD should closely match that with MSD (red solid circle) when multistage decoding is used in layer 1. Simulations in Figure 3.14 confirm this. Based on the increased coding rate, we are more concerned with the SER performance of layer 2. It is observed from Figure 3.14 that the SER performance of layer 2 is greatly degraded if MSD is not employed, with approximately 3 dB loss at 10^{-5} compared to the half-rate code used at this level. However, when MSD is used, the SER (blue solid square) of layer 2 has more than 3 dB gain over the non-MSD case (blue dashed square) as a result of the reliable *a priori* feedback from layer 1. The overall performance of MSD-based detection is determined mainly by layer 1, whereas for non-MSD-based detection, the overall performance is dominated by layer 2. That is the reason why the overall SER of the MSD-based scheme performs better than the non-MSD scenario, with 2 dB gain obtained at 10^{-5}. It is interesting to note that when the decoding of the Λ_{p_i}/Λ' which is constructed from a higher rate linear code occurs at a later stage of MSD, the overall SER performance of MSD over non-MSD performs better. Hence, MSD is particularly suitable for the detection of EDC lattices in terms of MLNC design, since each layer of EDC operates over an asymmetric finite field or finite chain ring. Now the overall SER is 4.5 dB from the capacity. Note that the measure of SER is based on the correct recovery of the linear combinations of original messages at each source over the respective algebraic field.

Iterative multistage decoding: We believe that there is room to improve SER and FER performance further. Based on the soft detector developed in Section 3.2.5, and also the soft decoder developed for the non-binary convolutional codes, we propose to apply the iterative technique to EDC-lattice-based MLNC.

Figure 3.14 depicts the result when IMSD is used. It is observed that with 5 iterations, the SER curve (black solid thick line) has a sharp turbo cliff reaching $\text{SER} = 10^{-5}$ at 10 dB, which is only 1.4 dB from the capacity. Thus, iterative decoding gives 3.3 dB gain over the traditional MSD decoding, and 5.3 dB gain over non-MSD decoding, as shown in the figure. Note that the simulation result is well consistent with the EXIT functions in Figure 3.12(a). When sufficient iterations are given, the L-value outputs from the soft detector at both layers are sufficiently reliable that the decoder can make the estimation with small probability of error. The simulation result also validates the soft detector algorithm specifically developed for EDC-based MLNC, and implies that there is large potential in employing iterative decoding in the multilevel lattice network coding.

Layered Integer Forcing: We have presented a general framework for the multilevel lattice network coding in Section 3.2.3. The work implies that any lattices with multilevel structure can be used in MLNC, and the essence of MLNC is to decode each layer separately such that the lattice decoder at each layer operates over smaller finite field or chain ring. The layered integer forcing is a network decoding technique developed in terms of

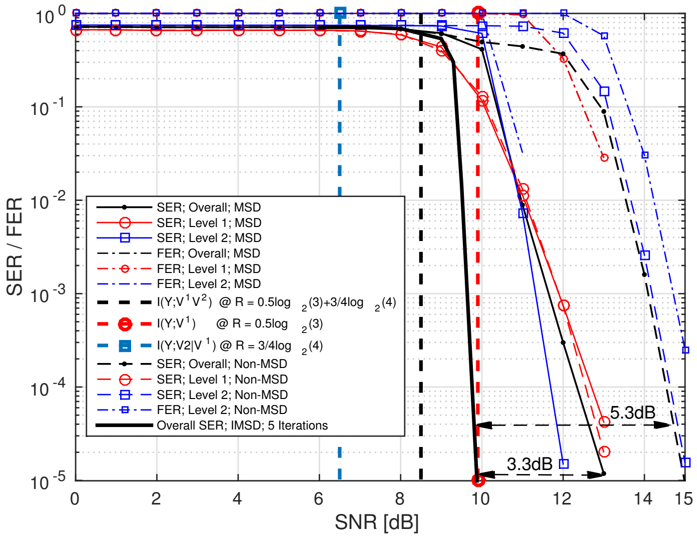


Figure 3.14: SER and FER performance for an MLNC constructed from a two layer EDC lattices; Soft detection; Multistage decoding/Non-multistage decoding/IMSD; Asymmetric coding rate; $\mathcal{R}_{\text{mes}}^{(1)} = \frac{1}{2}\log_2(3)$; $\mathcal{R}_{\text{mes}}^{(2)} = \frac{3}{4}\log_2(4)$; $h_1 = h_2 = 1$.

the algebraic structure of MLNC and hence, is generally applicable to any MLNC design. Thus, LIF is in principle capable of decoding EDC-lattice-based MLNC. According to Theorem 3.3, each layer forms a new quotient S-lattice Λ/Λ_i', and there exists a surjective S-module homomorphism φ_i for the i^{th} layer such that $\mathcal{K}(\varphi_i) = \Lambda_i'$. The general form of the generator matrix for Λ_i' based on the EDC lattice is given in (3.70). Note that Λ_i' is the coarse lattice for the new coset system Λ/Λ_i'.

In order to implement the LIF decoding for EDC lattices, we develop a modified Viterbi detector (see Section 3.2.8) which can be viewed as a lattice quantiser based on the quotient S-lattice Λ/Λ_i' for the i^{th} layer, $i = 1, 2, \cdots, m$.

Figure 3.15 illustrates the SER and FER performance based on LIF. It is observed (black solid line) that the overall SER has a good slope which validates the correctness of LIF and the modified Viterbi quantiser designed for the EDC lattice. The SER performance of LIF has approximately 0.8 dB loss at 10^{-5} in comparison to the soft detection approach. This is what we anticipate. First, the soft detection approach employs the BCJR algorithm for the convolutional decoding, which typically slightly outperforms Viterbi detection. Then, the soft detector developed in Section 3.2.5 and (3.96) – (3.99) outputs the soft information that the BCJR decoder uses to produce more reliable estimation than that for the Viterbi decoder.

The soft detection approach is designed specifically for EDC lattices, and it is not strange to see that it gives better performance than LIF. Despite of this, we emphasise that LIF

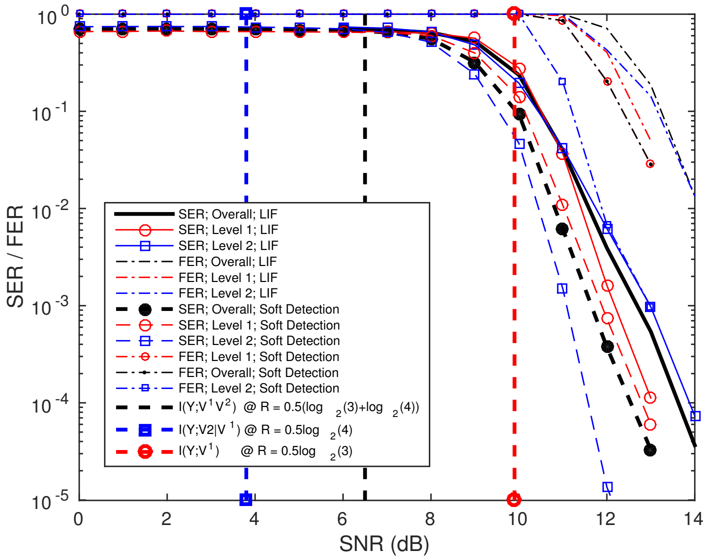


Figure 3.15: SER and FER performance for an MLNC constructed from a two layer EDC lattices; Soft detection; LIF; $\mathcal{R}_{\text{mes}}^{(1)} = \frac{1}{2}\log_2(3)$; $\mathcal{R}_{\text{mes}}^{(2)} = \frac{1}{2}\log_2(4)$; $h_1 = h_2 = 1$.

is universally applicable to any lattices having multilevel structure as detailed in Section 3.2.3, rather than just EDC lattices. For example, LIF is capable of solving MLNC problem when the lattices are directly designed in the Euclidean space, e.g. LDLC and signal codes. In summary, the application of the soft detection approach is more restrictive (which applies only to EDC lattices) and has relatively large complexity, but gives the best performance compared to LIF with Viterbi detection. However, LIF provides a solution for any kind of MLNC problem. Which method is preferable depends on the trade-off of factors relevant to a particular scenario.

In Figure 3.16, we also show the performance of the LSD when the fixed fading is considered. The channel fading vector is set to $\mathbf{h} = [-1.17 + 2.15i, 1.25 - 1.63i]$, which is the same as the fading vector used in scenario 1 of [74]. We employ a half-rate code for layer 1, and $\frac{3}{4}$-rate code for layer 2. The optimal S-integer vector for the two layers are selected in terms of (3.95). We employ multistage decoding with 5 iterations between the two layers. A sharp turbo cliff occurs, which reaches $\text{SER} = 10^{-5}$ at 3.9 dB, approximately 1.7 dB from the capacity. When no iteration is employed, there is more than 5 dB loss. This implies that small number of additional iterations to generate more reliable values is worthwhile in improving the overall SER performance. The iterative multistage soft detection for EDC lattices achieves the overall rate of $\mathcal{R}_{\text{mes}} \approx 2.29$ bits/symbol at 3.9 dB. This demonstrates the potential of iterative decoding in improving the performance of physical layer network coding.

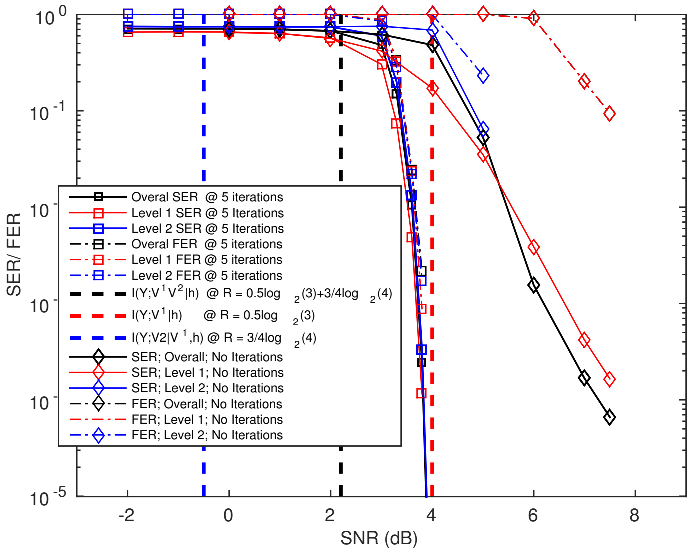


Figure 3.16: SER and FER performance for an MLNC constructed from a two layer EDC lattices; Soft detection; LIF; $\mathcal{R}^{(1)}_{\text{mes}} = \frac{1}{2}\log_2(3)$; $\mathcal{R}^{(2)}_{\text{mes}} = \frac{1}{2}\log_2(4)$; $\mathbf{h} = [-1.17 + 2.15i, 1.25 - 1.63i]$.

3.2.7 Conclusions

The paper has laid the foundations for a new research area in multilevel lattices for LNC, and built on the theoretic work for MLNC which inherently allows practically feasible decoding design for network coding, and correspondingly we have developed a layered integer forcing approach which plays such a role. We have proposed a general lattice construction, i.e. EDC, based on MLNC theorems, given the generator matrix forms and shown its merits, especially for complexity reduction and code design flexibility. We have considered three possible EDC lattice structures, and mathematically proved that EDC subsumes the most important previous complex constructions, e.g. A and D. We have laid the foundations for another new research area in iteration-aided multistage decoding for EDC-based MLNC, which is based on the layered soft detector developed in Section 3.2.5, and have explored its extrinsic information transfer characteristics. The results well support our viewpoint that LSD works well with multistage iterative decoding in MLNC, and provides better performance than the traditional non-iterative system. We have developed a modified Viterbi decoder based on LIF for EDC-based MLNC, and made performance comparison between iterative decoding, multistage decoding and LIF.

We expect that all of these will provide the basis for extensive further work, both to explore the rich algebraic features of the new construction and the homomorphism design, and to exploit it in practical implementations of LNC in 5G wireless systems.

3.2.8 Appendices

Layered soft detector for EDC-based MLNC

We show here the calculation of mutual information/ conditional mutual information between the received superimposed signals (faded and noisy) and the network coded symbols at the j^{th} level. We denote by $\mathbf{w}_j = [w_j^1, \cdots, w_j^m]$, $j = 1, 2, \cdots, L$ the realisations of a vector-based random variable $\mathbf{W}_j$ representing the messages of all m levels at the j^{th} source, and $\mathbf{w}^i = [w_1^i, \cdots, w_L^i]$, $j = 1, 2, \cdots, m$ the realisations of a vector-based random variable W^i representing the messages of all L sources for the i^{th} level. We refer to $\mathbf{w} = [\mathbf{w}_1, \cdots, \mathbf{w}_L]$ as the realisations of another vector-based random variable $\mathbf{W}$. Note that $w_j^i \in \mathbb{F}_{\tilde{p}_i}$ is the message of the i^{th} level and the j^{th} source, which is uniformly distributed over $\mathbb{F}_{\tilde{p}_i}$. V^i is a random variable which takes on a set of possible values of $f^i(w_1^i, \cdots, w_L^i) \in \mathbb{F}_{\tilde{p}_i}$, and v^i is the corresponding realisation of V^i for the i^{th} layer.

The *a posteriori* probability of the event $V^i = v^i$ at the i^{th} level conditioned on the MAC outputs $Y = y$ and the *a priori* channel outputs $\Xi = \boldsymbol{\xi} = [\xi^1, \cdots, \xi^m]$, is given by

$$\begin{aligned}
\Pr(V^i = v^i|y, \boldsymbol{\xi}) &= \sum_{V^i=v^i} \sum_{\mathbf{V}^{\backslash i} \in \mathbb{F}_{\mathbf{q}_{\backslash i}}^{m-1}} \frac{\Pr(Y|V^1 \cdots V^m, \boldsymbol{\xi}) \Pr(\boldsymbol{\xi}|V^1 \cdots V^m) \Pr(V^1 \cdots V^m)}{\Pr(Y, \boldsymbol{\xi})} \\
&= \Pr(\xi^i|V^i = v) \\
&\quad \sum_{V^i=v} \sum_{\mathbf{V}^{\backslash i} \in \mathbb{F}_{\mathbf{q}_{\backslash i}}^{m-1}} \frac{\Pr(Y|V^1 \cdots V^m) \Pr(\boldsymbol{\xi}^{\backslash i}|\boldsymbol{V}^{\backslash i}) \Pr(V^1 \cdots V^m)}{\Pr(Y, \boldsymbol{\xi})} \\
&= \frac{\Pr(\xi^i|V^i = v)}{\Pr(Y, \boldsymbol{\xi})} \sum_{\substack{\forall (w_1^i, \cdots, w_L^i): \\ f^i(w_1^i, \cdots, w_L^i) = v}} \sum_{\forall \boldsymbol{V}^{\backslash i} \in \mathbb{F}_{\mathbf{q}_{\backslash i}}} \sum_{\substack{\forall (\mathbf{w}_1^{\backslash i}, \cdots, \mathbf{w}_L^{\backslash i}): \\ f^1(\mathbf{w}^1) = v^1, \cdots, f^m(\mathbf{w}^m) = v^m}} \\
&\quad \Pr(Y|\mathbf{W} = \mathbf{w}) \Pr(\boldsymbol{\xi}^{\backslash i}|\boldsymbol{V}^{\backslash i}) \Pr(\mathbf{W} = \mathbf{w})
\end{aligned} \tag{3.96}$$

where $\mathbb{F}_{\mathbf{p}_{\backslash i}}^{m-1} \triangleq [\mathbb{F}_{\tilde{p}_1} \cdots \mathbb{F}_{\tilde{p}_{i-1}}, \mathbb{F}_{\tilde{p}_{i+1}} \cdots \mathbb{F}_{\tilde{p}_m}]$ consists of a set of finite field and finite chain ring. $\mathbb{F}_{\tilde{p}_i}$ is defined in Section 3.2.4. Note that if $\tilde{p}_i$ is not a prime number, $\mathbb{F}_{\tilde{p}_i}$ can be decomposed furthermore in terms of the p-adic decomposition theorem [64], and small modifications of (3.96) are required accordingly. The conditional probability density function is given by,

$$\begin{aligned}
\Pr(Y|\mathbf{W} = \mathbf{w}) &= \Pr(Y|\mathbf{w}_1, \cdots, \mathbf{w}_L) \\
&= \frac{1}{\sqrt{\pi N_0}} e^{-\frac{\left|h_1 \sigma^{-1}(\mathbf{w}_1^1 \times \cdots \times \mathbf{w}_1^m) + \cdots + h_L \sigma^{-1}(\mathbf{w}_L^1, \times \cdots \times \mathbf{w}_L^m) - y\right|^2}{N_0}}
\end{aligned} \tag{3.97}$$

The *a posteriori* L-value d_k^i for the event $V^i = v_k^i$ is defined in (3.98) which can be further separated into two terms in (3.99), where a_k^i is an element in (3.93) which serves as the *a priori* L-value. Note that v_k^i is the k^{th} realisation of the random variable V^i. Following (3.93), we have

$$\Pr(\xi|V = v_k^i) = \frac{e^{a_k^i}}{1 + \sum_{k:\forall v_k^i \in \mathbb{F}_{\tilde{p}_i}, v_k^i \neq 0} e^{a_k^i}}$$

The second term of (3.99) serves as the extrinsic L-value e_k^i for the i^{th} level and the k^{th} realisation of the vector-based random variable $\mathbf{E}^i$ which is the extrinsic information of the i^{th} level.

$$\begin{aligned} d_k^i &= \log\left(\frac{\Pr(V^i = v_k^i|y, \boldsymbol{\xi})}{\Pr(V^i = 0|y, \boldsymbol{\xi})}\right) && (3.98) \\ &= a_k^i + e_k^i, && (3.99) \end{aligned}$$

where

$$e_k^i = \log\left(\frac{\sum_{\substack{\forall(w_1^i,\cdots,w_L^i):\\ f^i(w_1^i,\cdots,w_L^i)=v_k^i}} \sum_{\forall \boldsymbol{V}^{\backslash i} \in \mathbb{F}_{\mathbf{q}_{\backslash i}}} \sum_{\substack{\forall(\mathbf{w}_1^{\backslash i},\cdots,\mathbf{w}_L^{\backslash i}):\\ f^1(\mathbf{w}^1)=v^1,\cdots,f^m(\mathbf{w}^m)=v^m}} \Pr(Y|\mathbf{W}=\mathbf{w})\Pr(\boldsymbol{\xi}^{\backslash i}|\boldsymbol{V}^{\backslash i})}{\sum_{\substack{\forall(w_1^i,\cdots,w_L^i):\\ f^i(w_1^i,\cdots,w_L^i)=0}} \sum_{\forall \boldsymbol{V}^{\backslash i} \in \mathbb{F}_{\mathbf{q}_{\backslash i}}} \sum_{\substack{\forall(\mathbf{w}_1^{\backslash i},\cdots,\mathbf{w}_L^{\backslash i}):\\ f^1(\mathbf{w}^1)=v^1,\cdots,f^m(\mathbf{w}^m)=v^m}} \Pr(Y|\mathbf{W}=\mathbf{w})\Pr(\boldsymbol{\xi}^{\backslash i}|\boldsymbol{V}^{\backslash i})}\right).$$

Mutual information for linear combinations

The mutual information between the received signal and the decoded linear combination at the i^{th} layer is:

$$\begin{aligned} I(Y,H;V^i) &= \boldsymbol{E}_{(Y,V^i,H)}\left[\log_2 \frac{P(Y|V^i,H)}{P(Y)}\right] \\ &= \sum_{v^i} \Pr(V^i = v_i) \int_{\mathbb{C}} P(H) \int_{\mathbb{C}} P(Y|V^i,H) \log_2 \frac{P(Y|V^i,H)}{P(Y)} \mathrm{d}Y\mathrm{d}H \end{aligned} \tag{3.100}$$

The probability density function $P(Y|V^i,H)$ conditioned on $V^i = v^i$ should be calculated by:

$$P(Y|V^i = v^i, H) = \frac{1}{\Pr(V^i = v^i)} \sum_{\forall \boldsymbol{v}^{\backslash i}} \sum_{\substack{\forall(\mathbf{w}_1,\cdots,\mathbf{w}_L):\\ f^1(\mathbf{w}^1)=v^1,\cdots,f^m(\mathbf{w}^m)=v^m}} P(Y|\mathbf{W}=\mathbf{w},h)P(\mathbf{W}=\mathbf{w}) \tag{3.101}$$

The conditional mutual information $I(Y;V^i|V^1\cdots V^{i-1})$ gives the maximum achievable rate at the i^{th} layer when the linear combinations of the preceding stages are perfectly known, which can be calculated by:

$$\begin{aligned} I(Y;V^i|V^1\cdots V^{i-1}) &= \boldsymbol{E}_{(Y,V^1,\cdots,V^i,H)}\left[\log_2 \frac{P(Y|V^1\cdots V^i,H)}{P(Y|V^1\cdots V^{i-1},H)}\right] \\ &= \sum_{v^1\cdots v^{i-1}} \Pr(V^1=v^1,\cdots,V^i=v_{i-1})\cdot \\ &\quad \sum_{v^i}\int_{\mathbb{C}} P(H)\int_{\mathbb{C}} P(Y,V^i|V^1\cdots V^{i-1},H)\cdot \\ &\quad \log_2\frac{P(Y|V^1\cdots V^i,H)}{P(Y|V^1\cdots V^{i-1},H)}\mathrm{d}Y\mathrm{d}H \end{aligned}$$

where the conditional probability density function $P(Y,V^i|V^1\cdots V^{i-1},H)$ should be calculated in terms of the random variables of the messages, which is given by:

$$\begin{aligned} &P(Y,V^i|V^1=v^1,\cdots,V^{i-1}=v^{i-1},H) \\ =&\frac{1}{\Pr(v^1,\cdots,v^{i-1})}\sum_{\forall\mathbf{v}\notin(v^1\cdots v^i)}\sum_{\substack{\forall(\mathbf{w}_1,\cdots,\mathbf{w}_L):\\ f^1(\mathbf{w}^1)=v^1,\cdots,f^m(\mathbf{w}^m)=v^m}} P(Y|\mathbf{W}=\mathbf{w},h)\Pr(\mathbf{W}=\mathbf{w}) \end{aligned} \tag{3.102}$$

where $P(Y|\mathbf{W}=\mathbf{w},h)$ is given in (3.97). Note that V^i, $i=1,2,\cdots,m$ is a random variable defined by the linear function of the i^{th} layer over $\mathbb{F}_{\tilde{p}_i}$. Every V^i operates over different finite filed or chain ring.

LIF quantiser

We show here a LIF quantiser $\mathcal{Q}_{\text{LIF}}^{(i)}$ implemented via a modified Viterbi decoder. The quantisation problem for the i^{th} layer can be mathematically expressed as:

$$\arg\min_{\mathbf{c}_i} ||\alpha^i\mathbf{y}-(\tilde{\sigma}^{-1}(\mathbf{c}^i)+\lambda_i')||^2 \tag{3.103}$$

$$=\arg\min_{\mathbf{c}_i} ||(\alpha^i\mathbf{y}-\tilde{\sigma}^{-1}(\mathbf{c}^i))-\mathcal{Q}_{\Lambda_i'}((\alpha^i\mathbf{y}-\tilde{\sigma}^{-1}(\mathbf{c}^i))||^2 \tag{3.104}$$

$$\text{subject to :}\qquad \mathbf{c}^i\in\mathcal{C}^i,\quad \lambda_i'\in\Lambda_i', \tag{3.105}$$

$$\tilde{\sigma}(\lambda)\in\mathcal{C}^1\oplus\cdots\oplus(\mathcal{C}^i=\mathbf{c}^i)\oplus\cdots\mathcal{C}^m \tag{3.106}$$

where $\mathcal{Q}_{\Lambda_i'}(\mathbf{x})$ is the coarse lattice quantiser for the i^{th} layer and can be expressed as a modulo operation $\mathbf{x} \bmod \Lambda_i'$ (as defined in Theorem 3.3). $\tilde{\sigma}^{-1}(\,\cdot\,)$ is the inverse operation of $\tilde{\sigma}$ which produces a set of lattice points λ.

We can construct trellis for the non-binary convolutional code $\mathcal{C}^i$. Assume that the states of the k^{th} and $(k+1)^{\text{th}}$ time slots are s_k and s_{k+1}, respectively. The codeword of the branch that exists from s_k and arrives at s_{k+1} is denoted as $c^i_{s_k \to s_{k+1}}$. The metric for each branch is given by

$$||(\alpha^i \mathbf{y} - \sigma^{-1}(c^i_{s_k \to s_{k+1}})) - \mathcal{Q}_{\Lambda'_i}((\alpha^i \mathbf{y} - \sigma^{-1}(c^i_{s_k \to s_{k+1}}))||^2 \tag{3.107}$$

where $\sigma^{-1}(\,\cdot\,)$ is the inverse operation of $\sigma(\,\cdot\,)$ defined in Section 3.2.4. We employ Viterbi algorithm to estimate the best possible outcome $\mathbf{c}^i$. This implements the LIF quantiser $\mathcal{Q}^{(i)}_{\mathrm{LIF}}$ for EDC-based MLNC.

Proof of Proposition 3.1 and Proposition 3.2

The codeword of the i^{th} layer is $\mathbf{c}^i = (c^i_1 + \langle\varpi\rangle, \cdots, c^i_n + \langle\varpi\rangle) \in \mathcal{C}^i$. $\mathcal{C}^i$ is a linear code over $\delta_i S/\langle\varpi\rangle$ which is generated by $\tilde{\sigma}_i([\mathbf{I}_{k_i} \;\; \mathbf{B}^i_{k_i \times (n-k_i)}])$ where $[\mathbf{I}_{k_i} \;\; \mathbf{B}^i_{k_i \times (n-k_i)}]$ is a $k_i \times n$ matrix over $\delta_i S$. These are defined in (3.64) and (3.65). The minimum-norm coset leader in the i^{th} layer primary sublattice system is given by:

$$\begin{aligned} \tilde{\sigma}_{i,\triangle}(\mathbf{c}^i) &= \big(c^i_1 - \mathcal{Q}_i(c^i_1/p_i\delta_i)p_i\delta_i, \cdots, c^i_n - \mathcal{Q}_i(c^i_n/p_i\delta_i)p_i\delta_i\big) \\ &= \big(c^i_1 - \varpi\mathcal{Q}_i(c^i_1/\varpi), \cdots, c^i_n - \varpi\mathcal{Q}_i(c^i_n/\varpi)\big) \end{aligned} \tag{3.108}$$

where $\mathcal{Q}(z)$ is a quantiser which sends $z \in \mathbb{C}$ to the closest point in S. We denote $d^2(\Lambda_{p_i}/\Lambda')$ as the length of the squared shortest vectors in the set $\Lambda_{p_i} \setminus \Lambda'$, then

$$d^2(\Lambda_{p_i}/\Lambda') = \min_{\mathbf{c}^i \neq \mathbf{0}, \mathbf{c}^i \in \mathcal{C}^i} ||\tilde{\sigma}_{i,\triangle}(\mathbf{c}^i)||^2 = \omega^{(i)}_{\min}(\mathcal{C}^i) \tag{3.109}$$

The volume of the Voronoi region of Λ' is $V(\mathcal{V}(\Lambda')) = \vartheta^n |\varpi|^{2n}$, where ϑ is a scaling factor depending on which PID is used, e,g., $\vartheta = \sqrt{3}/2$ when S is Eisenstein integer. The nominal coding gain for the i^{th} layer primary sublattices is:

$$\begin{aligned} \varrho(\Lambda_{p_i}/\Lambda') &= \frac{\omega^{(i)}_{\min}(\mathcal{C}^i)}{(V(\mathcal{V}(\Lambda_{p_i})))^{\frac{1}{n}}} \\ &= \frac{\omega^{(i)}_{\min}(\mathcal{C}^i)}{(\vartheta^n |p_i|^{2(n-k_i)} |\delta_i|^{2n})^{\frac{1}{n}}} \\ &= \frac{\omega^{(i)}_{\min}(\mathcal{C}^i)}{\vartheta |p_i|^{2(1-\frac{k_i}{n})} |\delta_i|^2} \end{aligned} \tag{3.110}$$

We now prove the kissing number for the i^{th} layer primary sublattices. Let $N(\omega^{(i)}_{\min}(\mathcal{C}^i))$ be the number of codewords in $\mathcal{C}^i$ with the minimum Euclidean weight $\omega^{(i)}_{\min}(\mathcal{C}^i)$, and $\mathcal{N}_{\mathcal{U}(S)}$ be the number of units in S, e.g. $\mathcal{N}_{\mathcal{U}(\mathbb{Z}([\omega]))} = 6$. When $|p_i|^2 - 1 \leq \mathcal{N}_{\mathcal{U}(S)}$, recall that $\mathcal{C}^i$ is a linear code over $\delta_i S/\langle\varpi\rangle$. The number of non-zero elements of coset leaders in $\delta_i S/\langle\varpi\rangle$ is $|p_i|^2 - 1$, and these elements have to be a subset of $\delta_i \mathcal{U}(S)$. Hence, there are $\frac{\mathcal{N}_{\mathcal{U}(S)}}{|p_i|^2 - 1}$ elements in the coset leaders which are formed by the same codeword and give the shortest vector. This means the number of the non-zero elements in a codeword is precisely $\frac{\omega_{\min}^{(i)}(\mathcal{C}^i)}{|\delta_i|^2}$. When $|p_i|^2 - 1 > \mathcal{N}_{\mathcal{U}(S)}$, every neighbour point is represented by different codewords, and hence the kissing number of the i^{th} layer primary sublattices is given by:

$$K(\Lambda_{p_i}/\Lambda') = \begin{cases} N(\omega_{\min}^{(i)}(\mathcal{C}^i)) \left(\frac{\mathcal{N}_{\mathcal{U}(S)}}{|p_i|^2-1}\right)^{\frac{\omega_{\min}^{(i)}(\mathcal{C}^i)}{|\delta_i|^2}}, & |p_i|^2 - 1 \leq \mathcal{N}_{\mathcal{U}(S)} \\ N(\omega_{\min}^{(i)}(\mathcal{C}^i)), & \text{Otherwise} \end{cases} \tag{3.111}$$

and now Proposition 3.1 is proved.

From the proof of Theorem 3.1, we have $\tilde{\mathbf{c}} = \mathbf{c}^1 + \mathbf{c}^2 + \cdots + \mathbf{c}^m$, $\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}$ and $\tilde{\mathcal{C}} \in (S/\langle \varpi \rangle)^n$. The minimum-norm coset leader for Λ/Λ' can be represented by codewords used for all layers, thus:

$$\tilde{\sigma}_{\triangle}(\tilde{\mathbf{c}}) = \left(\tilde{c}_1 - \varpi \mathcal{Q}\left(\frac{\tilde{c}_1}{\varpi}\right), \cdots, \tilde{c}_n - \varpi \mathcal{Q}\left(\frac{\tilde{c}_n}{\varpi}\right)\right) \tag{3.112}$$

where $\tilde{c}_j = c_j^1 + c_j^2 + \cdots + c_j^m$ and c_j^i, $j = 1, 2, \cdots, n$, $i = 1, 2, \cdots, m$, denotes the j^{th} element of the codeword $\mathbf{c}^i$. Then, the squared shortest vectors in the set $\Lambda \setminus \Lambda'$ can be represented by the m linear codes used at each layer,

$$d^2(\Lambda/\Lambda') = \min_{\tilde{\mathbf{c}} \neq \mathbf{0}, \tilde{\mathbf{c}} \in \mathcal{C}} ||\tilde{\sigma}_{\triangle}(\tilde{\mathbf{c}})||^2 = \omega_{\min}(\tilde{\mathcal{C}})$$

The nominal coding gain for Λ/Λ' is

$$\begin{aligned} \varrho(\Lambda/\Lambda') &= \frac{\omega_{\min}(\tilde{\mathcal{C}})}{(V(\mathcal{V}(\Lambda)))^{\frac{1}{n}}} \\ &= \frac{\omega_{\min}(\tilde{\mathcal{C}}) \prod_{\ell=2}^{m} |p_\ell|^{\frac{2(k_\ell - k_1)}{n}}}{\vartheta |p_1|^{2(1-\frac{k_1}{n})} |\delta_1|^2} \end{aligned} \tag{3.113}$$

where we assume $k_1 \leq k_2 \leq \cdots \leq k_m$ in (3.113). Let $N(\omega_{\min}(\tilde{\mathcal{C}}))$ be the number of codewords in $\tilde{\mathcal{C}}$ with the minimum Euclidean weight $\omega_{\min}(\tilde{\mathcal{C}})$, the kissing number of this kind of lattices is:

$$K(\Lambda/\Lambda') = N(\omega_{\min}(\tilde{\mathcal{C}})) \tag{3.114}$$

Proposition 3.2 is thereby proved.

Proof of Proposition 3.3 and Proposition 3.4

As explained in Section 3.2.4, the codeword $\mathcal{C}^i$ of the i^{th} layer operates over the finite chain ring $\mathcal{C}^i \in \delta_i S/\langle \varpi \rangle$, where $\delta_i = \varpi/p_i^{\gamma_i}$. Following (3.71), $\tilde{\sigma}_i(\mathbf{w}^i \mathbf{G}_{\text{FCR}}^i) \longmapsto (\delta_i S/\langle \varpi \rangle)^n$, here the message space of the i^{th} layer is defined as:

$$\mathbf{W}^i \cong (\delta_i S/p_i^{\gamma_i} \delta_i S)^{k'_{i,0}} \oplus \cdots \oplus (\delta_i S/p_i \delta_i S)^{k'_{i,\gamma_i - 1}}$$

Then the minimum-norm coset leader of Λ_{p_i}/Λ' has similar form as (3.108) with $\mathbf{c}^i \in \mathcal{C}^i$. The nominal coding gain $\varrho(\Lambda_{p_i}/\Lambda')$ can be obtained based on the same derivation in (3.109) and (3.110). We are more interested in constructing the primary sublattices with some linear codes over the finite field. This can be implemented via the complex construction D approach, based on a set of nested linear codes, as proved in Section 3.2.4. The residue field is now defined as $Q \triangleq \delta_i S/\langle p_i\delta_i\rangle$. Let $\mathcal{C}^{i,0} \subseteq \cdots \subseteq \mathcal{C}^{i,\gamma_i-1}$ be nested linear codes of length n over Q, where $\mathcal{C}^{i,t}$ is an $[n, \sum_{\ell=0}^{t} k'_{i,\ell}]$ linear code for the t^{th} level of the i^{th} layer, $t = 0, 1, \cdots, \gamma_i - 1$. Note that $\mathcal{C}^{i,t}$ is row spanned by the vector space:

$$\mathbf{g}_{\mathcal{C}^{i,t}} = \begin{bmatrix} \mathbf{g}^i_{k'_{i,0}} \\ \mathbf{g}^i_{k'_{i,1}} \\ \vdots \\ \mathbf{g}^i_{k'_{i,t}} \end{bmatrix} \tag{3.115}$$

where $\mathbf{g}^i_{k'_{i,t}} \in Q_{k'_{i,t}\times n}$. None of the rows of $\mathbf{g}_{\mathcal{C}^{i,t}}$ are linear combination of the other rows. It is obvious that the primary sublattice point $\lambda_{p_i} \in \Lambda_{p_i} \setminus \Lambda'$ is given by:

$$\lambda_{p_i} = \underbrace{\underbrace{\underbrace{p_i^{\gamma_i}\delta_i\mathbf{s} + p_i^{\gamma_i-1}\mathbf{c}^{i,\gamma_i-1}}_{\Lambda_{p_i}^{\gamma_i-1}} + \cdots + p_i\mathbf{c}^{i,1}}_{\Lambda_{p_i}^{1}} + \mathbf{c}^{i,0}}_{\Lambda_{p_i}^{0}} \tag{3.116}$$

where $\mathbf{c}^{i,t}$ has to not be zero for all possible t values. The outer lattice $\Lambda_{p_i}^{\gamma_i-1} = \{p_i^{\gamma_i-1}(p_i\delta_i\mathbf{s} + \mathbf{c}^{i,\gamma_i-1}) = p_i^{\gamma_i-1}\Lambda^{\perp}_{i,\gamma_i-1} : \mathbf{s} \in S^n, \mathbf{c}^{i,\gamma_i-1} \in Q^n \setminus \mathbf{0}\}$. Here $\Lambda^{\perp}_{i,\gamma_i-1}$ forms a lattice which has the same structure as the one in scenario 1, with $\Lambda^{\perp'}_{i,\gamma_i-1} = \{p_i\delta_i\mathbf{s} : \mathbf{s} \in S^n\}$. Thus, $\mathbf{c}^{i,t} = (c_1^{i,t} + \langle p_i\delta_i\rangle, \cdots, c_n^{i,t} + \langle p_i\delta_i\rangle) \in \mathcal{C}^{i,t}$. The minimum-norm coset leader for $\Lambda^{\perp}_{i,\gamma_i-1}/\Lambda^{\perp'}_{i,\gamma_i-1}$ and the minimum Euclidean weight $\omega_{\min}^{(i,\gamma_i-1)}(\mathcal{C}^{i,\gamma_i-1})$ can be readily obtained in the same way as (3.108) and (3.109). It is obvious that $\lambda_{p_i}^{\gamma_i-1} \in p_i^{\gamma_i-1}\Lambda^{\perp}_{i,\gamma_i-1} \setminus p_i^{\gamma_i-1}\Lambda^{\perp'}_{i,\gamma_i-1}$ and we have $\| \lambda_{p_i}^{\gamma_i-1} \|^2 \geq |p_i^{\gamma_i-1}|^2\omega_{\min}^{(i,\gamma_i-1)}(\mathcal{C}^{i,\gamma_i-1})$. The squared shortest vectors of the inner lattice, e.g. $\| \lambda_{p_i}^0 \|^2$ has to be at least larger than $\| \lambda^{\perp}_{i,0} \|^2$ where $\Lambda^{\perp}_{i,0} \triangleq \{\lambda^{\perp}_{i,0} = p_i\delta_i\mathbf{s} + \mathbf{c}^{i,0} : \mathbf{s} \in S^n, \mathbf{c}^{i,0} \in Q^n \setminus \mathbf{0}\}$, and we have $\| \lambda_{p_i}^0 \|^2 \geq \omega_{\min}^{(i,0)}(\mathcal{C}^{i,0})$. The squared shortest vectors in the set $\Lambda_{p_i} \setminus \Lambda'$ is therefore lower bounded by

$$d^2(\Lambda_{p_i}/\Lambda') \geq \min_{0\leq t\leq\gamma_i-1}\{|p_i|^{2t}\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})\}$$

where $\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})$ is referred to as the minimum Euclidean weight of non-zero codewords in $\mathcal{C}^{i,t} \in Q^n$ (an $[n, \sum_{\ell=0}^{t} k'_{i,\ell}]$ linear code) for the t^{th} level of the i^{th} layer. The nominal coding gain for the i^{th} layer primary sublattices in scenario 2 is lower bounded by:

$$\begin{aligned}\varrho(\Lambda_{p_i}/\Lambda') &\geq \frac{\min_{0\leq t\leq \gamma_i-1}\{|p_i|^{2t}\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})\}}{(V(\mathcal{V}(\Lambda_{p_i})))^{\frac{1}{n}}} \\ &= \frac{|p_i|^{2\sum_{t=0}^{\gamma_i-1}(\gamma_i-t)\frac{k'_{i,t}}{n}}\min_{0\leq t\leq \gamma_i-1}\{|p_i|^{2t}\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})\}}{\vartheta|\varpi|^2}\end{aligned} \tag{3.117}$$

Let $N_t(\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t}))$ be the number of codewords in $\mathcal{C}^{i,t}$ with the minimum Euclidean weight $\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})$ for the t^{th} level and the i^{th} layer. The kissing number of the i^{th} layer primary sublattice is upper bounded by

$$K(\Lambda_{p_i}/\Lambda') \leq \begin{cases} \sum_{t=0}^{\gamma_i-1} N_t(\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t}))\left(\frac{\mathcal{N}_{\mathcal{U}(S)}}{|p_i|^2-1}\right)^{\frac{\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})}{|\delta_i|^2}}, & |p_i|^2-1\leq \mathcal{N}_{\mathcal{U}(S)} \\ \sum_{t=0}^{\gamma_i-1} N(\omega_{\min}^{(i,t)}(\mathcal{C}^{i,t})), & \text{Otherwise} \end{cases} \tag{3.118}$$

This completes the proof of Proposition 3.3.

We now define the code $\mathcal{C}^i$ such that $\mathbf{c}^i = \mathbf{c}^{i,0} + p_i\mathbf{c}^{i,1} + \cdots + p_i^{\gamma_i-1}\mathbf{c}^{i,\gamma_i-1}$, and hence $\mathcal{C}^i \in \delta_i S/\langle\varpi\rangle$. From the theorems developed in sections 3.2.3 and 3.2.4, we are able to generate a new code $\tilde{\mathcal{C}}$ such that $\tilde{\mathbf{c}} = \mathbf{c}^1 + \cdots + \mathbf{c}^m$ which makes $\tilde{\mathcal{C}} \in S/\langle\varpi\rangle$. Thus, the codeword of $\tilde{\mathcal{C}}$ is generated by the nested linear codes $\mathcal{C}^{i,t}$ of all layers. The minimum-norm coset leader for Λ/Λ' can be represented by:

$$\tilde{\sigma}_{\triangle}(\tilde{\mathbf{c}}) = \left(\tilde{c}_1 - \varpi\mathcal{Q}\left(\frac{\tilde{c}_1}{\varpi}\right), \cdots, \tilde{c}_n - \varpi\mathcal{Q}\left(\frac{\tilde{c}_n}{\varpi}\right)\right) \tag{3.119}$$

where $\tilde{c}_j = c_j^1 + c_j^2 + \cdots + c_j^m$ and c_j^i, $j = 1, 2, \cdots, n$, $i = 1, 2, \cdots, m$, denotes the j^{th} element of the codeword $\mathbf{c}^i$. Then, the squared shortest vectors in the set $\Lambda \setminus \Lambda'$ can be represented by the m linear codes used at each layer,

$$d^2(\Lambda/\Lambda') = \min_{\tilde{\mathbf{c}}\neq\mathbf{0},\tilde{\mathbf{c}}\in\mathcal{C}} ||\tilde{\sigma}_{\triangle}(\tilde{\mathbf{c}})||^2 = \omega_{\min}(\tilde{\mathcal{C}})$$

The nominal coding gain for Λ/Λ' is

$$\begin{aligned}\varrho(\Lambda/\Lambda') &= \frac{\omega_{\min}(\tilde{\mathcal{C}})}{(V(\mathcal{V}(\Lambda)))^{\frac{1}{n}}} \\ &= \frac{\omega_{\min}(\tilde{\mathcal{C}})\prod_{\ell=1}^{m}|p_\ell|^{2\sum_{t=0}^{\gamma_\ell-1}(\gamma_\ell-t)\frac{k'_{i,t}}{n}}}{\vartheta|\varpi|^2}\end{aligned} \tag{3.120}$$

The nominal coding gain of the EDC lattice in scenario 2 is related to the minimum Euclidean weight of the composite code $\tilde{\mathcal{C}}$ and the code rates of all nested linear codes at each layer.

4 HNC maps and node operations in large scale scenarios

WPLNC based communication networks deliver the information from sources to destinations through the complex relay network. In order to properly design the Network Coded Modulation (NCM) transmitted by network nodes and the relay processing including the Hierarchical Network Code (HNC) maps, we need a formal description of the global network processing function. Any practical implementation also implies the half-duplex constrained relays which imposes the network to work in multiple stages. We present a polynomial based formalism defining the Hierarchical Network Transfer Function (H-NTF). It captures all phenomena related to the stage dependent transmit and receive activity over the network, including potential buffering, and mainly the hierarchy of forming the local HNC maps into the global hierarchical flow description. Using the polynomial formalism of H-NTF, we develop the half-duplex constrained stage scheduling algorithm. It starts with finding a causal minimal latency (or close to minimal) critical sequence with subsequent doubly (first Rx then Tx) greedy mapping of the node activity compliant with the half-duplex constraint.

4.1 Hierarchical network transfer function

4.1.1 Introduction

WPLNC based communication networks deliver the information from sources to destinations through the complex relay network. Each node demodulates, decodes, processes and re-encodes the hierarchical information (many-to-one function of the component data) directly in the constellation space. Various aspects of the design ranging from the NCM (in many flavors: Compute and Forward, Denoising, Hierarchical Decode and Forward, etc.) over the relay node strategies (decode, compress) to the Hierarchical Information (HI) and Hierarchical Side Information (HSI) decoding strategies are discussed in number of works, e.g. [22, 75–79].

This paper deals with a specific problem of WPLNC design related to the *half-duplex stage scheduling*. In order to properly design the NCM transmitted by network nodes and the relay processing (including the Hierarchical Network Code (HNC) maps), we need a formal description of the global network processing function. Any practical implementation also implies the half-duplex constrained relays which imposes the network to work in multiple stages.

It appears that all works on WPLNC assume either quite simplistic (e.g. two-way relaying) or at least a priori given and fixed stage scheduling. At the best, the given fixed scheduling is parameterised by adjusting the time proportions of the stages. The fixed scheduling is then followed by either purely ad-hoc choice of HNC maps or some sort of the optimisation but only inside a very limited constraints of the selected stages scheduling, e.g. compute-and-forward requires "layered" type of scheduling. The uniting *generic model* capable of describing the hierarchy of the WPLNC processing for *arbitrarily complicated network* is still missing. Also the generally applicable *synthesis* of the stage scheduling in complicated topologies is unsolved. The scheduling of network resources is, indeed, a wide and deeply investigated area of the theory of networks. Its main target is however rather aimed on data/packet flows in the network. It should *not* be confused with network *stage* scheduling. It is rather physical layer related and defines how the network should be split into its per-state (stage) defined sub-networks and what type of activity are nodes supposed to do and how the local edge/node properties/behaviour unfolds into the global network description.

The background theory frequently used to solve the tasks of global network description is the graph theory, e.g. [80]. The graph theory gives the global view and properties of the network (the graph) based on its structure description typically represented by the adjacency matrix. The local edge or node properties are represented by scalars only. They represent node adjacency by 0 or 1, or are extended to scalar integer or real values describing the multiplicity of capacity of the edges. This local description is however insufficient for describing complex behaviour of WPLNC networks. On the other side, the coding theory uses polynomial formalism frequently for the analysis of trellis free path properties of codes that can be described by the finite state machine, e.g. convolutional and turbo codes [43]. A specific situation when the network (graph) changes in time appears to have received only limited attention, and if so, it still remains within a perspective of simple scalar local description on the edges. The work [81] introduces the graph unfolding but still tightly bound to scalar edge description, in this particular case even using Boolean operations.

This paper builds on these ideas and develops a network graph model that can handle (i) complex local description of the edge and node properties using polynomial formalism, and, at the same time, (ii) allows for time dependent network graph with discrete ordered causal states (stages). The local and per-stage description is unfolded into the global description capturing the whole hierarchy of local properties. The model is then used to synthesise half-duplex constrained stage scheduling. The *motivation and goals* are outlined here.

(1) The design of a WPLNC network involves the synthesis and optimisation of relay strategies that process many-to-one functions (hierarchical information) of the data streams represented in the constellation space. At the same time, we have to guarantee the end-to-end solvability for the desired data at the destination based on the observation of multiple hierarchical information carrying signals. Having a description connecting the local and the global behaviour is a first step in this design.

(2) Let us assume that the building blocks of the WPLNC cloud are *half-duplex constrained* relays performing their front-end strategy (multiple stage Hierarchical Multiple Access Channel (H-MAC)), back-end strategy (Hierarchical Broadcast Channel (H-BC)) and relay processing strategy described by HNC map χ and it corresponding processing metric μ. We are also given a connectivity map of the network.

(3) The goal is to develop a technique that will allow to build the whole *encapsulation hierarchy* of the information flow between the source and its target destination respecting all involved HNC maps, Tx activity stages, received signals participating in a given HNC map, mixed stages flows and potential buffering at nodes.

The paper provides the following *contributions and results.*

(1) We develop a polynomial formalism describing all node activities (involved HNC maps, Tx activity stages, participating Rx signals, buffering). This will be used to define *Hierarchical Network Transfer Function (H-NTF)* capturing the whole encapsulation of these phenomena in the network. It can be used to identify the end-to-end solvability for the information flow including all hierarchical encapsulations, scheduling of the stages and it will help identifying the critical bottlenecks. Since complex networks usually have a high diversity potential and also the processing and scheduling provide many possible options, it will also set the model for *optimising* the node operations. The name "hierarchical" emphasises that the NTF captures the encapsulation *hierarchy* rather than the input-output transfer [82].

(2) Using the H-NTF formalism, we develop the *doubly greedy half-duplex constrained stage scheduling* algorithm. It starts with finding a minimal latency (or close to minimal) causal critical sequence with subsequent doubly (first Rx then Tx) greedy mapping of the node activity compliant with half-duplex constraint.

4.1.2 Polynomial formalism

Assume WPLNC cloud network with nodes numbered by integers $\mathcal{S} = \{1, \ldots, K\}$. Sources and destinations are included in this set. Let the set of source nodes indices be $\mathcal{S}_S = \{i_1, \ldots, i_{K_S}\} \subset \mathcal{S}$ and corresponding (correctly ordered) destination nodes $\mathcal{S}_D = \{\hat{i}_1, \ldots, \hat{i}_{K_S}\} \subset \mathcal{S}$. Indices in $\mathcal{S}_D$ can repeat if a given node is a destination for multiple sources. We also (additionally over the numerical indices) denote the sources and destinations with letter indices, $S_{i_1} \equiv S_A, S_{i_2} \equiv S_B \ldots$ and $D_{\hat{i}_1} = D_A, D_{\hat{i}_2} = D_B, \ldots$. Nodes participating in ℓ-th stage have indices from the set $\mathcal{S}_\ell \subset \mathcal{S}$, $\ell \in \{1, \ldots, L\}$.

A directed connectivity $K \times K$ matrix for ℓ-th stage is $\mathbf{H}_\ell$ where columns correspond to Tx activity and rows to Rx activity. Its i-th row j-th column entries $H_{\ell,ij}$ are equal to 1 if i-th node receives the signal from j-node in the ℓ-th stage, otherwise they are 0. Notice, that the connectivity matrix under the half-duplex constraint has zeros on the main diagonal. We also define a global directed connectivity matrix $\mathbf{H}_0$ which describes the connectivity *regardless of the stage* and its entries are $H_{0,ij} = \mathcal{U}\left(\sum_{\ell=1}^{L} H_{\ell,ij}\right)$, where $\mathcal{U}(\boldsymbol{.})$

is a unit step function. Entries are equal to 1 if any of the per-stage directed connectivity element is 1.

For each stage, we define per-stage $K \times K$ network transfer matrix

$$\mathbf{G}_\ell = W_\ell \left(\mathbf{X}_\ell \mathbf{H}_\ell \mathbf{V}_\ell + \mathbf{B}_\ell \right) \tag{4.1}$$

. The polynomial formalism denotes the passing of the network flow through ℓ-th stage by W_ℓ. The $K \times K$ diagonal matrix $\mathbf{X}_\ell$ represents the event of the network flow passing through the i-th receiver's HNC map in ℓ-th stage by the polynomial variable $X_{\ell,i}$, i.e. $\mathbf{X}_\ell = \operatorname{diag}[X_{\ell,1}, X_{\ell,2}, \ldots, X_{\ell,K}]$ where $\operatorname{diag}[a_1, \ldots, a_n]$ denotes a diagonal matrix with $a_1, \ldots, a_n$ on main diagonal. Similarly, the $K \times K$ diagonal matrix represents the transmit activity of j-th node in the ℓ-th stage by the polynomial variable $V_{\ell,j}$, i.e. $\mathbf{V}_\ell = \operatorname{diag}[V_{\ell,1}, V_{\ell,2}, \ldots, V_{\ell,K}]$. The diagonal $K \times K$ matrix $\mathbf{B}_\ell$ represents the buffering at j-th node at stage ℓ. It adds diagonal entries on $\mathbf{G}_\ell$ matrix, i.e. the node virtually receives what it has transmitted $\mathbf{B}_\ell = \operatorname{diag}[B_{\ell,1}, B_{\ell,2}, \ldots, B_{\ell,K}]$. In summary, the matrix $\mathbf{G}_\ell$ has nonzero entries (1) on main diagonal $W_\ell B_{\ell,j}$ which represent buffering, and (2) on i-th row and j-th column $W_\ell X_{\ell,i} V_{\ell,j}$ if node i receives the signal from node j in ℓ-th stage.

4.1.3 Hierarchical network transfer function

The Hierarchical Network Transfer Matrix (H-NTM) is defined as a compound network transfer matrix over all stages[1]

$$\mathbf{F} = \sum_{\ell=1}^{L} \prod_{m=1}^{\ell} \mathbf{G}_m = \mathbf{G}_1 + \mathbf{G}_2\mathbf{G}_1 + \cdots + \mathbf{G}_L\mathbf{G}_{L-1} \ldots \mathbf{G}_2\mathbf{G}_1. \tag{4.2}$$

The Hierarchical Network Transfer Function (H-NTF) is multi-stage network response $\mathbf{z}$ on the excitation from the sources $\mathbf{s}$ evaluated at the proper destination indices $\mathbf{z} = [\tilde{z}_{\hat{i}_1}, \ldots, \tilde{z}_{\hat{i}_{K_S}}]^T$ where destination nodes have indices $\mathcal{S}_D = \{\hat{i}_1, \ldots, \hat{i}_{K_S}\} \subset \mathcal{S}$, the full response for all nodes is $\tilde{\mathbf{z}} = \mathbf{F}\mathbf{s}$ and $\mathbf{s}$ is the source excitation vector with entries S_i on positions $i \in \mathcal{S}_S = \{i_1, \ldots, i_{K_S}\} \subset \mathcal{S}$ otherwise zeros.

Hierarchical network transfer matrix is the network response combining the results from all stages, e.g. $\mathbf{G}_1\mathbf{s}$ is the response after first stage with source excitation $\mathbf{s}$, $\mathbf{G}_2\mathbf{G}_1\mathbf{s}$ is the response after first and second stage, etc. The H-NTF contains a complex information about the network flows. It can be simplified and interpreted in various was as we explain later. The example of the polynomial formalism and H-NTF is in Figure 4.1.

[1] This follows similar ideas as [81] unfolding accessibility principle but with generalised polynomial entries. Also, in the per-stage network transfer matrix product, the subsequent states (the later stages) are left-hand matrix multiplications. We however do not use an explicit notation for this in the product operator. We also assume a finite number L of stages and no pipelining over several repetitions of L-stage sequence.

The complete H-NTF $\mathbf{z}$ contains complex information about the data flow in the WPLNC network. For specific purposes, we can *simplify* it in order to reveal what we want. Also we can *optimise* the cloud operation by selectively switching on/off various functionalities and subsequently analysing the result.

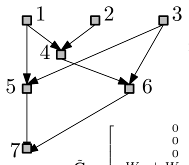
The most important aspect is whether the desired source found its way to the intended destination. This is fulfilled if the source appears in the corresponding destination H-NTF. However it, on its own, does not guarantee the solvability (generalised exclusive law), it is only a necessary condition. Buffering can be switched off by simply setting all $\tilde{B} = 0$. We can also do that selectively for individual nodes and/or stages. We define zero-buffering H-NTF $\mathbf{z}_{\bar{B}} = \mathbf{z}|_{\tilde{B}=0}$. The hierarchical encapsulation of the WPLNC network is revealed by identifying the HNC maps at particular nodes and particular stages where the source participates in. For this purpose we evaluate $\mathbf{z}_{\bar{B},X} = \mathbf{z}_{\bar{B}}|_{\tilde{W}=1,\tilde{V}=1}$. If the network has high hierarchical information path diversity, we can selectively switch off some nodes' transmission by setting $V_{\ell,j} = 0$ for a given ℓ, j, i.e. $\mathbf{z}_{\bar{B},X,\bar{V}_{\ell,j}} = \mathbf{z}_{\bar{B}}|_{\tilde{W}=1,\tilde{V}=1,V_{\ell,j}=0}$.

In the network with high diversity of the end-to-end flows having a non-uniform number of stage activity over the paths, we need a tool to recognise the role of signals in terms of HI and HSI. Components (or sub-components) of H-NTF can be HI or HSI only if they have the same number of W_ℓ variables. It indicates that the source data come from the same epoch (defined by the stages) and therefore have a chance to support themselves (HI) or help resolve a friendly interference (HSI). Otherwise the data comes from different epochs and they are independent (if the source data are IID).

4.2 Doubly greedy stage scheduling

The polynomial formalism of the network transfer matrices can be used for the *design* of the half-duplex constrained scheduling. There are many possible half-duplex scheduling possibilities. Apart of the half-duplex constraint, we impose additional requirements to reduce the number of possible solutions. A natural additional requirement is to minimise the *latency* while keeping the multi-stage data flow *causal.*

The forthcoming algorithm solves the half-duplexing systematically while the latency and causality is solved by enforcing an ad-hoc solution which, however, in many case gives the minimum latency solution. Essentially, we will identify the per-stage network transfer matrices $\mathbf{G}_\ell$ fulfilling the half-duplex constraint while enforcing the critical transmission sequence. It guarantees that the data flow on the critical path (typically the longest path) will causally find its way to the desired destination with minimal latency. The algorithm is doubly greedy in a sense that (1) all receivers that can hear transmitters on critical path are set to the reception mode on a given stage, (2) all transmitters that do not violate the half-duplex constraint (dictated by the previous point) are allowed to transmit. We can later switch them off selectively after analysing and optimising the



$$\mathbf{H}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \mathbf{G}_{\mathrm{cr}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & W_3 & 0 \end{bmatrix} \mathbf{v}_{\mathrm{hd}} = \left(\textstyle\sum_{\ell=1}^{L} W_\ell\right)\mathbf{i} - \mathbf{r}_{\mathrm{cr}} = \begin{bmatrix} W_1 + W_2 + W_3 \\ W_1 + W_2 + W_3 \\ W_1 + W_2 + W_3 \\ W_2 + W_3 \\ W_1 + W_2 + W_3 \\ W_1 + W_3 \\ W_1 + W_2 \end{bmatrix}$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ W_1 + W_2 + W_3 & W_1 + W_2 + W_3 & 0 & 0 & 0 & 0 & 0 \\ W_1 + W_2 + W_3 & 0 & W_1 + W_2 + W_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & W_1 + W_2 + W_3 & W_2 + W_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & W_1 + W_2 + W_3 & W_1 + W_3 & 0 \end{bmatrix} \quad \mathbf{r}_{\mathrm{cr}} = \mathbf{G}_{\mathrm{cr}}\mathbf{i} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ W_1 \\ 0 \\ W_2 \\ W_3 \end{bmatrix}$$

$$\begin{aligned} z_{ABC} = {} & S_A W_1 W_2 \big(W_3 B_{1,1} V_{2,1} V_{3,5} X_{2,5} X_{3,7} + V_{1,1} \big(\left(1 + W_3 B_{3,7}\right) V_{2,5} X_{1,5} X_{2,7} + W_3 \left(B_{2,5} V_{3,5} X_{1,5} + V_{2,4} V_{3,6} X_{1,4} X_{2,6} \right) X_{3,7} \big) \big) \\ & + S_B W_1 W_2 W_3 V_{1,2} V_{2,4} V_{3,6} X_{1,4} X_{2,6} X_{3,7} \\ & + S_C W_1 W_2 \big(W_3 B_{1,3} V_{2,3} \left(V_{3,5} X_{2,5} + V_{3,6} X_{2,6} \right) X_{3,7} + V_{1,3} \big(\left(1 + W_3 B_{3,7}\right) V_{2,5} X_{1,5} X_{2,7} + W_3 \left(B_{2,5} V_{3,5} X_{1,5} + B_{2,6} V_{3,6} X_{1,6} \right) X_{3,7} \big) \big) \end{aligned}$$

Figure 4.1: 3-source 3-relay 1-destination example network. Sources are $\mathcal{S}_S = \{1, 2, 3\}$, $\mathbf{s} = [S_1, S_2, S_3, 0, 0, 0, 0]^T$, destinations are $\mathcal{S}_D = \{7, 7, 7\}$, $S_1 \equiv S_A, S_2 \equiv S_B, S_3 \equiv S_C$, $D_7 \equiv D_A, D_7 \equiv D_B, D_7 \equiv D_C$. H-NTF at final destination is z_{ABC}.

H-NTF. Putting first the greedy reception before the greedy transmission attempts to minimise the number of interacting signals in the WPLNC cloud.

Half-duplex constrained stage Rx-Tx greedy scheduling algorithm with *enforced latency-critical causal sequence* is given by the following steps.

1. Global directed connectivity. The node global radio visibility is defined by global directed connectivity matrix $\mathbf{H}_0$ and we assume that it is known.

2. Minimum latency causal path. We identify the minimum latency causal path. It is the longest directed and sequentially numbered path in the network graph between any of the sources and their corresponding destinations. Thus, it is the minimum number of hops if we respect only the directed connectivity regardless of the half-duplex constraint. This can be obtained by observing the source flow propagation through the network with increasing number of the hops. We observe the response $(\mathbf{H}_0\mathbf{V}_m) \times \cdots \times (\mathbf{H}_0\mathbf{V}_2)(\mathbf{H}_0\mathbf{V}_1)\mathbf{s}$ with sequentially increasing $m = 1, 2, \ldots$. The smallest m (denoted by $M_{\min}$) such that all sources find their way at least for some $m \leq M_{\min}$ to their corresponding destinations becomes the longest path ensuring causal delivery of source flow to the destination. The sequential multi-hop and causality principle also guarantees that the nodes on one individual path of given source flow (ignoring other sources and paths) are consistent with half-duplex constraint. The corresponding ordered set $\mathcal{S}_{\min}$ of transmitting nodes can be easily identified from the set of variables $\{V_{1,i_1}, \ldots, V_{m,i_m}\}$ associated with given source variable S_i. It will be the *minimum latency causal path* and it defines *mandatory* transmit activity of the nodes. If there are multiple of them, we choose randomly (or the one which is better for WPLNC used on top of the stage scheduling) one and cross-check the end-to-end flow for all sources in the following steps. If it fails, we choose a different one until all options are exploited. The complexity of this step is $\mathcal{O}(K^2 M_{\min})$.

3. Critical sequence. The previous step however does *not* generally guarantee, when we later impose the half-duplex constraint, that the all other sources find their way to their destinations in the number of half-duplex hops limited by $M_{\min}$. If this happens, we have to ad-hoc choose another enforced and possibly longer sequence of the transmitting nodes (not violating the half-duplex) and cross-check that the subsequent half-duplex schedule guarantees the end-to-end flow. This sequence $\mathcal{S}_c$ of transmitting nodes will be called *enforced latency-critical causal sequence*, or simply the *critical sequence*. The critical sequence guarantees the minimum latency causal network if the step #2 succeeded. Otherwise the minimum latency network does not exist and the critical path becomes ad-hoc solution. However, if the minimum latency path from the step #2 remains a subset of the critical path we get a solution which is close to the minimum latency one.

4. Mapping the critical sequence on stages. The critical sequence of transmitting nodes $\mathcal{S}_c = \{m_1, m_2, \ldots, m_L\}$ defines the stages. The node m_ℓ belongs to the stage $\ell \in \{1, \ldots, L\}$. It means that the node m_ℓ mandatory (by the critical sequence) transmits in stage ℓ. This mandatory transmission is represented by multiplying m_ℓ-th column (corresponding to the m_ℓ-th Tx activity) of the matrix $\mathbf{H}_0$ by the stage variable W_ℓ. The critical sequence transfer matrix is $\mathbf{G}_{\text{cr}} = \mathbf{H}_0 \operatorname{diag}[w_1, w_2, \ldots, w_K]$ where $w_i = W_\ell$ if $m_\ell = i$ otherwise $w_i = 0, i \in \{1, \ldots, K\}, \ell \in \{1, \ldots, L\}$. Columns of $\mathbf{G}_{\text{cr}}$ that belong to mandatory transmissions are labeled by the corresponding W_ℓ. Columns which do not participate in mandatory critical sequence transmission are set to zero. Since the critical sequence was set as a causal Tx activity sequentially mapped on the stages, each stage appears only once in the matrix $\mathbf{G}_{\text{cr}}$.

5. Critical sequence Rx nodes (greedy Rx). All nodes that can receive the signals from the critical sequence are set to the receive mode in the corresponding stage. These nodes, regardless whether they are on the critical path, i.e. greedy Rx, can be found by evaluating $\mathbf{r}_{\text{cr}} = \mathbf{G}_{\text{cr}}\mathbf{i}$ where $\mathbf{i} = [1, 1, \ldots, 1]^T$. The i-th component of $\mathbf{r}_{\text{cr}}$ contains the sum of variables W_ℓ of the stages received by the i-th receiver. The complexity of this step is $\mathcal{O}(KL)$.

6. Half-duplex constrained Tx (greedy Tx). Nodes that do not receive in the given stage are allowed to transmit in that stage (greedy Tx). The set of allowed transmission half-duplex stages is simply get by subtracting (in polynomial representation) the reception vector $\mathbf{r}_{\text{cr}}$ from the vector containing all stages $\mathbf{v}_{\text{hd}} = \left(\sum_{\ell=1}^{L} W_\ell\right) \mathbf{i} - \mathbf{r}_{\text{cr}}$. The allowed Tx stages are then mapped onto the *half-duplex transfer generating matrix* $\tilde{\mathbf{G}} = \mathbf{H}_0 \operatorname{diag}(\mathbf{v}_{\text{hd}})$. The generating matrix has nonzero entries on the positions inherited from the directed global connectivity matrix $\mathbf{H}_0$ and each nonzero entry is a sum of W_ℓ variables representing the half-duplex consistent allowed Tx stages. The complexity of this step is $\mathcal{O}(KL)$. The half-duplex generating matrix *only* guarantees that the *critical* *path* nodes comply with the half-duplex constraint and otherwise it gives doubly-greedy freedom of Rx/Tx which can but does not have to be utilised.

7. Per-stage connectivity matrices. Per-stage directed connectivity matrix $\mathbf{H}_\ell$ is simply get by taking the generator matrix and setting $W_{\ell'} = 0$ for all stages $\ell' \neq \ell$ $\mathbf{H}_\ell = \tilde{\mathbf{G}}|_{W_\ell=1,\, W_{\ell'}=0 \text{ for } \ell' \neq \ell}$.

Algorithm needs to performed by the centralised entity with $\mathbf{H}_0$ knowledge and then subsequently the nodes are informed.

4.2.1 Example application.

The half-duplex Rx-Tx greedy scheduling procedure with enforced latency-critical causal path is demonstrated on example network in Figure 4.1. Minimum latency causal path is obtained from evaluating $(\mathbf{H}_0\mathbf{V}_m) \times \cdots \times (\mathbf{H}_0\mathbf{V}_2)(\mathbf{H}_0\mathbf{V}_1)\mathbf{s}$ and observing the response at node 7, i.e. the 7-th component of the responses

$$[(\mathbf{H}_0\mathbf{V}_2)(\mathbf{H}_0\mathbf{V}_1)\mathbf{s}]_7 = S_1 V_{1,1} V_{2,5} + S_3 \left(V_{1,3} V_{2,5} + V_{1,3} V_{2,6}\right), \tag{4.3}$$

$$[(\mathbf{H}_0\mathbf{V}_3)(\mathbf{H}_0\mathbf{V}_2)(\mathbf{H}_0\mathbf{V}_1)\mathbf{s}]_7 = S_1 V_{1,1} V_{2,4} V_{3,6} + S_2 V_{1,2} V_{2,4} V_{3,6}. \tag{4.4}$$

We see that S_1, S_3 reach the destination in two steps while S_2 needs three steps. Then the minimum latency causal path (and also the critical sequence) is dictated by S_2 and it is $\mathcal{S}_c = \{2, 4, 6\}$ (there is no other option) and the number of stages is $L = 3$. The cross-check of the end-to-end flow is simplified by the fact that all source have a common destination and therefore only one component z_{ABC} of H-NTF needs to evaluated. All three sources reach the destination and we can use the H-NTF to analyse and optimise stage activity, buffering and HNC maps.

4.3 Cloud access node scheduling and validation

A large body of research has been recently devoted to the problem of wireless multi-user communications. Traditional approaches of multi-hop relay networking use the orthogonal separation of resources. However, in dense multi-relay ad-hoc networks these approaches scale poorly with network size due to the increase of the overall interference [83]. Recently the dense WCN architecture has been proposed [84–87] as a promising paradigm for multi-user communications based on the use of intermediate relay nodes. The key feature of this architecture is low latency and high reliable multi-user communication enabled by interference mitigation techniques at the PHY layer of the relay nodes. This is referred as cloud-relaying and it is a cluster of densely deployed relay nodes with the same ad-hoc properties. In order to accelerate the proliferation

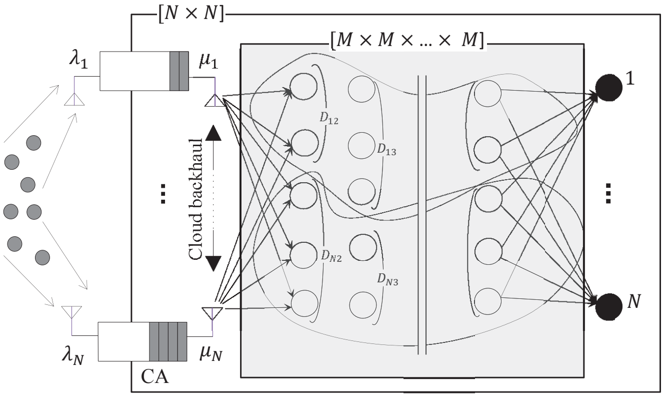


Figure 4.2: Super dense wireless cloud network (WCN) reference architecture.

of such a high throughput and reliable communications consistent with 5G vision the quest for some required technologies is started in terms of: (i) introducing novel system architectures and resource allocation schemes, (ii) advance transmission techniques and strategies, (iii) new relays or new functions in current relays.

In the WCN, transmitted signals from multiple flows are flooded from the source CA nodes to destinations through densely deployed relay nodes, in a layered architecture as shown in Figure 4.2. Relay nodes (generally decode or amplify or compute and forward relays) are cooperating directly at the PHY layer, creating a compound virtual macro-relay that employs advanced interference processing [85]. The fundamental goal of the relay nodes is to deal with the interference of simultaneous flows by adopting interference processing techniques such as interference neutralisation [88], wireless network coding [11], and successive interference cancellation [89]. The performance of the dense WCN strictly depends on the way multiple flows coexist within the WCN. Moreover, intrinsic heterogeneity of the arrival traffic, caused by the mobility pattern of end-users, is another dominant factor in CA nodes binary (ON/OFF) mode. While some CA nodes are inactive (OFF mode) due to the lack of traffic, other CA nodes might be overwhelmed with a high burst of traffic. Despite of the fact that WCN can perfectly mitigate the interference impairment and accommodate all the CA nodes in active (ON) mode, this barely happens due to the heterogeneity of arrival rates. Since the WCN capacity scales with the number of simultaneously active CA nodes, this leads to waste resources and decrease the cloud efficiency in handling multiple flows in parallel.

The purpose of this section is the design of CA nodes scheduling mechanism and an offloading mechanism in the backhaul of the WCN. CA node scheduling is inspired by the adoption of the scheduling approach proposed in [90]. Two optimal scheduling solutions, so-called *one-at-a-time* and *all-at-once* modes and their general optimality conditions are discussed in [90]. Intuitively, *one-at-a-time* is optimal if the transmitted flows pose heavy interference to each other, such that the rate reduction due to interference is significant. In fact, *one-at-a-time* is the cloud mode when baseline TDMA is the optimal transmission scheme. Conversely, *all-at-once* mode is preferable for mitigated-interference scenario, where simultaneous transmissions have virtually no impact on each others' transmission rate. Considering the interference mitigation capability of the dense WCN, *all-at-once* mode is the preferable scheduling solution for the CA nodes in the WCN if interfering signals are perfectly canceled by the WCN's internal processing. We further enhance this scheduling mechanism through an ad-hoc offloading mechanism devised in the backhaul of the WCN (among CA nodes) to maximise its utility, i.e. multiple flows transmission with the minimal cloud duty-cycle. The proposed scheme is evaluated via Matlab and also simulations at the System Level Simulator (SLS).

4.3.1 System model

The WCN consists of a set of N CA nodes acting as $N \times N$ MIMO system toward N destinations through a macro-relay node. We have L layers from source to destination CA nodes. The macro-relay is a cascade of $[M \times M \times \cdots \times M]$ MIMO system with $L-2$ layers and M relays in each layer, equipped with specific interference mitigation capability that can provide parallel orthogonalised communication channels for the active flows (up to N) originated from CA nodes. Notice that $N \leq M$ and each flow can be relayed with one or more relay nodes in each layer.

Basically, in each layer l, relay nodes are assigned to the transmission flows to provide a robust virtual communication tunnel for each flow. We define $\mathcal{D}_{il}$ as relay set consisting of relays (here decode and forward) dedicated to the i-th transmission flow at the l-th layer. All relays in $\mathcal{D}_{il}$ receive their signal from i-th flow, decode (or denoise [11]) and forward it to the next layer, while treating the signals from other flows as noise that are assumed to be mitigated through functionality enabled at the PHY. This internal layered architecture provides redundant copies of desired signals in each layer, resulting reliable communication links between source and destination CA nodes [91]. Obviously, flows with more stringent QoS requirements will be allocated with larger number of relays in each layer.

Packets originated from geographically distributed clients with a Poisson-Like arrival rates $\{\lambda_i\}_{i=1}^{N}$, accumulated in finite size queues at CA nodes. Each CA node transmits the packets stored in its queue through the WCN to the destination CA nodes. The service rates experienced by CA nodes $\{\mu_i\}_{i=1}^{N}$, as offered by the WCN, are random and exponentially distributed with a rate that depends on the state of the cloud s and the internal processing gain α. Cloud state s refers to the number of simultaneously activated CA nodes (ON mode), and the internal processing gain $\alpha \in [0,1]$ depicts the ratio of the interfering signals that reach the destination after being suppressed by the interference mitigation capability of the WCN. Just to exemplify, if $\alpha = 0$ the WCN mitigates all the interference impairments in relaying multiple flows, while for $\alpha = 1$ the WCN has no interference mitigation capability. When the interfering signals are perfectly mitigated, more CA nodes can transmit simultaneously via cloud. Potentially, in each transmission round, up to N CA nodes can simultaneously transmit their queued packets to N receiver CA nodes (Namely cloud acts in "all-at-once mode, providing for a point to point communication model through orthogonalised channels as for $\alpha = 0$). cloud utilisation factor is defined as $(\frac{T_{tdma}}{T_{cloud}})$, where T_{tdma} is the entire transmission time of all the CA nodes in the baseline time-division-multiple-access scheme (TDMA) where one CA node at a time is using the WCN in a Round Robin (RR) approach, and T_{cloud} is the entire transmission time of all the CA nodes via WCN (namely, WCN duty-cycle).

We define W_{cloud} and W_b as portions of the total bandwidth allocated to the WCN and its backhaul communication respectively. More in detail, cloud bandwidth W_{cloud} is used for the source to destination CA nodes communication through the WCN, and the backhaul bandwidth W_b is used for the backhaul communication so that CA nodes can exchange their queue state information and offload their backlogged packets to each other (if beneficial). Basically, the use of the WCN as a macro-relay with simultaneous transmissions is beneficial, if $T_{cloud} \leq T_{tdma}$. The scope of this section is designing the CA nodes scheduling and its enhancement through offloading in the cloud backhaul, assuming the WCN resources, e.g. cloud bandwidth W_{cloud}, backhaul bandwidth W_b and relay sets in each layer $\mathcal{D}_{il}$, are granted.

4.3.2 Cloud access node scheduling

In this subsection we address the CA nodes scheduling that is to find the minimum scheduling time for emptying all queued packets at all CA nodes conditioned to the incoming traffic $\{\lambda_i\}_{i=1}^N$, inter-CA nodes coordination and queues offloading. The objective of the WCN is simultaneous transmission of as much as flows compatible with the transmit capability of the WCN for a given internal interference mitigation gain α. Although CA node scheduling facilitates the simultaneous transmission, it does not necessarily guarantee *all-at-once* mode. In fact, the transmission rate per each flow is impaired by increasing the number of simultaneous flows, depending on the WCN interference processing gain α. Considering the fact that each CA node has a specific service rate granted by the cloud (which can be decreased when increasing the number of simultaneous CA nodes) finding the optimal sets of CA nodes for simultaneous transmissions with the minimum WCN time is the focus of this subsection. We model the minimum-time scheduling problem of CA nodes, and verify the effect of the interference processing gain and heterogeneity of traffic on the cloud states, in the following.

As introduced, let $\mathcal{N} = \{1, 2, \ldots, N\}$ be the set of CA nodes (or equivalently N flows) can be activated within t-th round of transmission, these CA nodes are associated with queues states $\mathbf{q} = \{q_1, q_2, \ldots, q_N\}$ where q_i represents the accumulated packets at the i-th CA node before the t-th round of transmission. The set of all possible scheduling subsets of $\mathcal{N}$ is $\mathcal{H}$ with cardinality $|\mathcal{H}| = 2^N - 1$, excluding the empty set (no usage of the cloud). We use the term state when referring to $s_h \in \mathcal{H},\ h = \{1, 2, \ldots, |\mathcal{H}|\}$, that is the subset of the CA nodes that are scheduled together (ON mode) with multiple flows.

The WCN in h-th state (s_h) means that all CA nodes in the subset s_h are in ON mode to transmit their encoded flows simultaneously. For any given state s_h, the rate of each CA node depends on the composition of CA nodes in s_h. Let $f(i, s_h)$ denote the rate function for a flow originated from i-th CA node when $i \in s_h$. These rates represent feasible rates that are within the capacity region granted by the cloud.

The minimum-time scheduling problem for a given (f,**q**), is to select a set of states $(s_1, s_2, \ldots, s_H)$, among the $2^N - 1$ members of $\mathcal{H}$, along with their respective allocated transmission times $(T_1, T_2, \ldots, T_H)$ such that the total time $\sum_{h=1}^{H} T_h$ is minimised, provided that the corresponding data rate $\{\{f(i, s_1)\}_{i=1}^{N}, \{f(i, s_2)\}_{i=1}^{N}, \ldots, \{f(i, s_H)\}_{i=1}^{N}\}$ can empty the queues **q**. Based on the above mentioned points we can write

$$\min_{T_{s_h}} \sum_{s_h \in \mathcal{H}} T_{s_h} \; , \tag{4.5}$$

subject to the constraints

$$\sum_{s_h \in \mathcal{H}} f(i, s_h) T_{s_h} = q_i, \;\; i = 1, \ldots, N \; , \tag{4.6a}$$

$$T_{s_h} \geq 0, \;\; h = 1, \ldots, |\mathcal{H}| \; . \tag{4.6b}$$

One optimal solution for (4.5) can be scheduling all N flows simultaneously, in *all-at-once* mode. If we consider s_H as the state including all the CA nodes then $f(i, s_H) \geq \frac{f(i,s_h)|_{|s_h|=1}}{N}$ is the necessary condition for the *all-at-once* solution, where $f(i, s_h)|_{|s_h|=1}$ denotes the service rate of i-th CA node in the baseline TDMA scheme. Note that, if s_h denotes the h-th state from $\mathcal{H}$ (or equivalently a subset of simultaneous CA nodes) and i is the index of a CA node so that $i \in s_h$, the service rate $f(i, s_h) \leq f(i, s_k)$, if $s_h \subset s_k$. It means that by increasing the number of simultaneous flows, throughput of each flow can gradually reduce, even if the total cloud throughput might be still good compared to the baseline TDMA scheme.

Cloud service rate: For the sake of reasoning, we approximate the service rate $f(.)$ from Signal-to-Noise-and-Interference Ratio (SINR) on links, cloud internal processing gain α, and the number of hops between source and destination CA nodes. Basically, the end-to-end service rate for i-th CA node in the h-th state of the cloud can be defined as

$$f(i, s_h) = \frac{W_{cloud}}{L-1} \log_2 \left(1 + \min_{r_l} \left\{ \frac{\sum_{r \in \mathcal{D}_{i,l-1}} p_\circ g_{r,r_l}}{\sigma_\eta^2 + \alpha \Gamma_l(i, s_h)} \right\} \right) , \tag{4.7}$$

where $p_\circ$ is the transmission power, g_{r,r_l} is the gain of Rayleigh channel for two successive relays (namely, $(r, r_l) \in \{\mathcal{D}_{i,l-1} \times \mathcal{D}_{i,l}\}$), and σ_η^2 is the noise term. $\Gamma_l(i, s_h)$ is the interference impairment caused by the other simultaneous flows at the l-th layer, that is scaled by cloud internal processing gain α. Note that the end-to-end service rate is scaled down by the number of hops ($L - 1$) according to the layered architecture of the cloud (see Figure 4.2) that allocates orthogonal resources at every hop as for other networks. The optimal number of layers L with the minimum outage probability is investigated in [92] with analysis restricted to one relay at each layer. A closed-form expression for the optimal number of layers incorporating the end-to-end throughput needs further investigation. Considering a fixed number of layers, cloud service rate $f(i, s_h)$ for i-th CA node in h-th state depends on the internal processing gain α which quantifies the efficiency of the cloud in attenuating the interference impairment. Note that for small value of α (when $\alpha \to 0$) the number of simultaneous transmissions increases (due to the perfect interference mitigation capability) while for large values of α (when $\alpha \to 1$) the CA nodes will be scheduled closer to the conventional TDMA or *one-at-a-time* mode. Finally, the average service rate of i-th CA node can be approximated as

$$\mu_i = \sum_{s_h \in \mathcal{H}} \frac{T_{s_h}}{T_{cloud}} f(i, s_h) \ . \tag{4.8}$$

In order to have a tractable analysis, the service rate for the i-th CA node is exponentially distributed with parameter μ_i.

4.3.3 Distributed offloading in the cloud backhaul

In this subsection we highlight that the heterogeneous arrival rate can degrade the performance of the cloud scheduler. In fact, even if the cloud sustains the maximum possible service rate by perfectly mitigating the interference impairment, the traffic pattern, i.e. the number of queued packets in each CA node, is a dominant factor in achieving the optimal solution. Specifically, the solution provided by (4.5) depends upon two factors: (i) the cloud service rate and (ii) CA nodes queue state. The former is a function of cloud internal processing gain characterised by parameter α (see subsection 4.3.2). The latter depends on the traffic rates that are not essentially homogeneous.

Figure 4.3 visualises two different solutions of scheduler (4.5), depicting the evolution of the queue state on the scheduling and consequently cloud states. As we already mentioned, if we assume s_H is the state that all CA nodes are activated in *all-at-once* mode, with cardinality $|s_H| = N$ and $f(i, s_H) \geq \frac{f(i,s_h)|_{|s_h|=1}}{N}$ for $\forall i \in s_H$, the state s_H is the optimal solution of (4.5). However, heterogeneous packet arrival rates can manipulate the optimal solution by reducing the cardinality of the activated CA nodes, for a period of time, increasing the cloud duty-cycle (T_{cloud}).

Despite the fact that the cloud internal processing gain can support the simultaneous transmission of all the CA node, the cloud state s_h strictly depends on the state of the queues $\mathbf{q}$. As is shown in Figure 4.3 a, when packet arrival rates of CA nodes are heterogeneous (here we consider $\lambda_3 > (\lambda_1, \lambda_2) > \lambda_4$ for a cloud with $N = 4$ CA nodes and one layer of relays, i.e. $L = 3$ hops), the scheduler first equalises the queue length by activating CA nodes in the groups with cardinality of two or three and finally activates

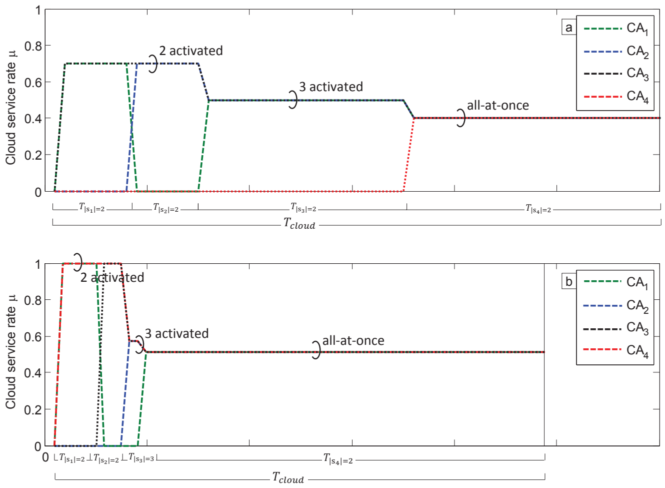


Figure 4.3: Cloud processing state s_h, depending on the queue states: a) heterogeneous packet arrival rates ($\lambda_3 > (\lambda_1, \lambda_2) > \lambda_4$); b) homogeneous packet arrival rates ($\lambda_i = \lambda_j \;\; \forall i, j \in \mathcal{N}$).

all of the CA nodes in *all-at-once* mode. On the other hand, when queues are almost balanced with homogeneous arrival rates ($q_i = q_j$ for $i, j \in N$) the cloud is activated in *all-at-once* mode in a larger portion of the cloud duty-cycle ($T_{(|s_h|=N)}/T_{cloud}$) such that the cloud is fully utilised, see Figure 4.3 b. Notice that the cloud is fully utilised if $T_{(|s_h|=N)}/T_{cloud} \to 1$, as a larger amount of traffic can be processed and reach the destination nodes in a shorter period of time.

We devise a distributed offloading mechanism in the backhaul of the cloud to equalise the number of the queued packets at the source CA nodes. Parallel to the transmission through the cloud, CA nodes partially use W_b to offload some queued packets from high-demanded CA nodes to the low-demanded ones. This scheme implicitly uses CA nodes as cooperative relay nodes in the backhaul of the cloud and accordingly forces the scheduler to activate all the CA nodes as they have approximately equalised queue size. We pair *donor* and *acceptor* CA nodes for cooperation, based on their communication delay in the cloud backhaul. Although every CA node can potentially communicate with all the other CA nodes, due to the scarcity of resources, each CA node constructs a group of three CA nodes to cooperate in the offloading process. Let r_{ji} be the normalised fraction of packets that i-th CA node forwards to the j-th CA node and $\rho_j = \mu_j^{'} - \sum_{k=1}^{N} r_{jk}\lambda_k$ be the residual service capability of j-th CA node. Note that $\mu_j^{'} = 1/(T_j^c + T_j^b)$ where T_j^c is the average serving time of a packet in the j-th CA node (waiting time in the queue plus transmission time) and T_j^b is the offloading delay in the backhaul of the cloud depending on the channel quality used in the cloud backhaul. In other words, $\mu_j^{'}$ is the service rate of the i-th CA nodes' packet when it is redirected to the j-th CA node. In compact notation, $\boldsymbol{\rho} = \boldsymbol{\mu}^{'} - \mathbf{R}\boldsymbol{\lambda}$ where $\mathbf{R}$ is the $N \times N$ offloading matrix among CA nodes. Each row $\mathbf{r}_j = [r_{j1}, r_{j2}, \ldots, r_{jN}]$ is the incoming portion of traffic toward j-th CA node and each column $\mathbf{r}_i = [r_{1i}, r_{2i}, \dots, r_{Ni}]^T$ is the outgoing portion of traffic from i-th CA node to the other CA nodes.

As mentioned earlier, we let every CA node offloads its packets toward two *acceptor* CA nodes with the minimum offloading time. In fact, each CA nodes finds the nodes with the lowest offloading delay in the cloud backhaul. Every CA node constructs the offloading set $\mathfrak{J} = \{i, l, k\}$ consists of its own index i and two CA nodes for offloading, indexed as l, k. The purpose here is optimising the column vector $\mathbf{r}_i$ of each CA node based on the knowledge of the residual service capability ρ_j for $j \in \mathfrak{J}$. Therefore, delay for each CA node for incoming traffic can be written as

$$D_i(\mathbf{R}) = \sum_{j=1}^{N} r_{ji}\tau_j(\mathbf{R}) \ , \tag{4.9}$$

where $\tau_j(\mathbf{R}) = \frac{1}{\rho_j - r_{ji}\lambda_i}$. Regarding the above mentioned parameters, each CA node minimises its cost function as

$$\min_{\boldsymbol{r}_i} D_i(\mathbf{R}) \ , \tag{4.10}$$

subject to the constraints:

$$\sum_{j=1}^{N} r_{ji} = 1 \ , \tag{4.11a}$$

$$r_{ji} \geq 0 \ \ \forall j \in \mathfrak{J} \ , \tag{4.11b}$$

$$r_{ji} = 0 \ \ \forall j \in \mathcal{N} \backslash \mathfrak{J} \ . \tag{4.11c}$$

Assuming that CA nodes in $\mathfrak{J}$ are sorted based on their available capacity ($\rho_1 \geq \rho_2 \geq \rho_3$), the optimal outgoing portion of backlogged packets from i-th CA node to the j-th CA node in $\mathfrak{J}$ is

$$r_{ji} = \begin{cases} \frac{1}{\lambda_i}\left(\rho_j - \frac{\sqrt{\rho_j}}{\sqrt{\gamma}}\right) & \text{if } 1 \leq j < I \\ 0 & \text{if } I \leq j \end{cases} \ , \tag{4.12}$$

where

$$\sqrt{\gamma} = \frac{\sum_{j=1}^{I-1} \sqrt{\rho_j}}{\sum_{j=1}^{I-1} \rho_j - \lambda_i} \ , \tag{4.13}$$

and I is the minimum index that satisfies the inequality

$$\sqrt{\rho_I} \leq \frac{1}{\sqrt{\gamma}} \ . \tag{4.14}$$

Proof. In Section 4.3.6.

According to (4.10) every CA node minimises its delay by direct transmission or offloading through the other CA nodes. This model implicitly selects neighbouring CA nodes,

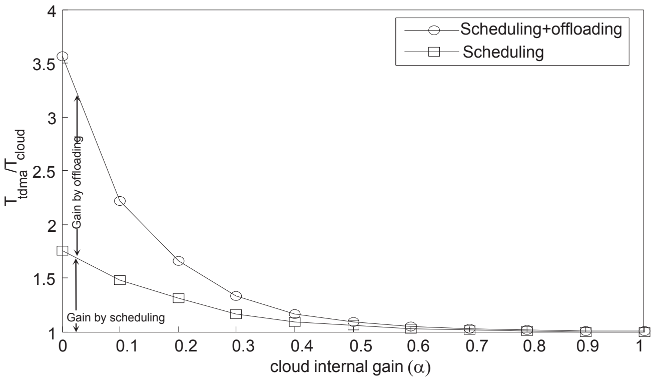


Figure 4.4: Cloud utilisation factor (T_{tdma}/T_{cloud}) versus cloud internal gain $\alpha \in [0,1]$ under heterogeneous arrival rates. $N = 4$.

or CA nodes with a high quality communication channel for offloading. Assuming that the service rate of the cloud is fair to all the CA nodes (α is equal for all flows) this offloading mechanism equalises queue sizes at the source CA nodes. We verify the proposed minimum-time scheduling and distributed offloading schemes with some numerical results in the following subsection.

4.3.4 Numerical results

In this subsection, we evaluate the performance of the proposed scheduling and distributed offloading mechanism compared to the conventional baseline TDMA scheme. Simulation is twofold. We use Matlab to evaluate the cloud utilisation factor when we use both minimum-time scheduling and offloading mechanisms. Then we use the SLS as a more realistic environment to evaluate end-users QoS, e.g. packet delay and packet loss ratio. The SLS is a flexible tool with a modular object-oriented software architecture, which allowed the rapid implementation and evaluation of the proposed schemes. It focuses on Layer-2 protocols, but it also implements the PHY functionalists, i.e. simulating the underlying wireless channel and the WCN interference mitigation capability.

Figure 4.4 compares the cloud utilisation factor $1 \leq \frac{T_{tdma}}{T_{cloud}} \leq N$ versus cloud internal processing gain $\alpha \in [0,1]$. For simplicity, $N = 4$ pairs of transmitter-receiver CA nodes are considered with heterogeneous arrival rates so that the packet arrival rate of the second CA node is two times higher compared to the others (say $\lambda_2 = 2\bar{\lambda}$ and $\lambda_i = \bar{\lambda}\ \forall i \neq 2$). As shown for $\alpha = 0$ (namely when cloud perfectly mitigates the interference) the minimum-time scheduler joint with distributed offloading mechanism in the cloud backhaul fully utilises the cloud under heterogeneous arrival rates and outperforms the minimum scheduling time solely. Moreover, there is no performance gain compared to the TDMA scheme when $\alpha \to 1$ as there is no interference mitigation capability which would facilitate parallel transmissions. Figure 4.5 highlights the effect of the cloud internal interference processing gain α on the dedicated time portion to each state of the

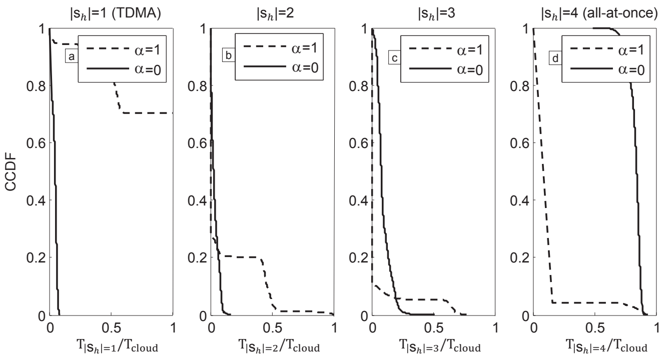


Figure 4.5: Complementary CDF ($\Pr[t \geq T_{|s_h|}/T_{cloud}]$) of time portions ($T_{|s_h|}/T_{cloud}$) with minimum-time scheduler and offloading mechanism under heterogeneous arrival rates with cloud internal gain $\alpha \in \{0, 1\}$.

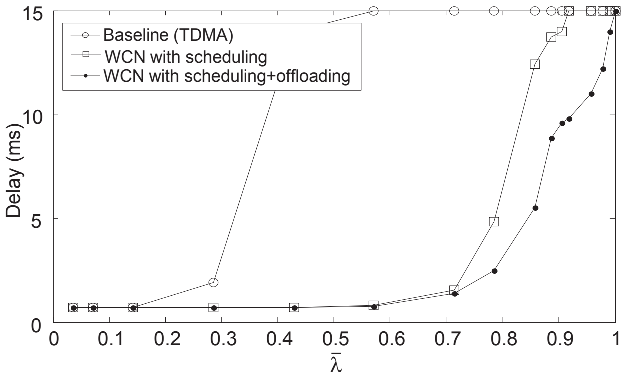


Figure 4.6: Comparison of packet delay between scheduler joint with offloading and TDMA scheme under different average load factors $\bar{\lambda}$, $N = 4$.

cloud, i.e. T_{s_h} as the time dedicated to the state s_h, based on the complementary CDF, under heterogeneous arrival rates. As is shown, cloud behaves differently depending on the value of α. Cloud is operating in TDMA mode (about 70% of the entire duty-cycle) when $\alpha = 1$, while the highest percentage of the cloud duty-cycle is spent on the *all-at-once* mode when $\alpha = 0$.

We evaluate the packet delay and packet loss ratio of the cloud equipped with the proposed schemes through the SLS. In all the simulations made by SLS maximum capacity per each flow is assumed 54 Mbps, but varying based on the SINR value. Traffic is generated with an exponential process, and stored in CA node buffers with a capacity of 50 packets, awaiting transmission. The mean rate of the exponential process is derived from the input load which is an independent parameter, and is expressed relative to the capacity. Unbalanced traffic is assumed in all our experiments, with even-numbered nodes assigned double traffic load that the odd-numbered ones. This is to test the effectiveness of the offloading mechanism. In Figure 4.6, 4.7 we compare the performance of the scheduling algorithm and offloading scheme, compared to the baseline TDMA scheme,

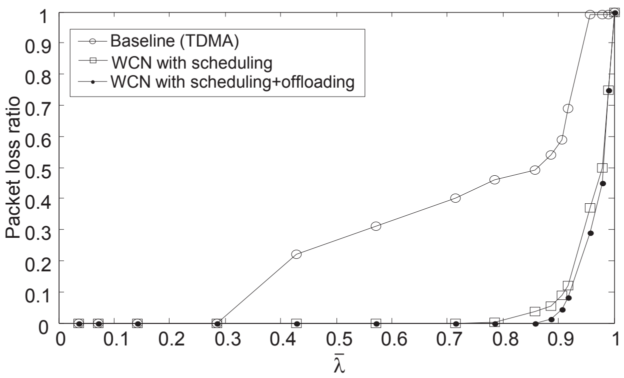


Figure 4.7: Comparison of packet loss ratio between scheduler joint with offloading and TDMA scheme under different average load factors $\bar{\lambda}$, $N = 4$.

i.e. round-robin transmission. It can be seen that the proposed schemes significantly outperform the baseline, as they allow multiple parallel transmissions. Additionally, it can be seen that the proposed offloading scheme further enhances performance, especially in high traffic loads. To begin with there is a clear reduction in the packet loss ratio, which is owed to the fact that without offloading in effect, unbalanced traffic saturates the wireless links that correspond to the even-numbered nodes, leaving the rest lightly loaded. Distributed offloading mechanism leads to more balanced traffic profiles and decreased packet loss. It also leads to lower packet delays, by keeping packet buffers better balanced. However, as can be seen, when the WCN saturates, i.e. $\bar{\lambda} \to 1$, offloading mechanism has no effect since all the queues are full.

4.3.5 Conclusion

In this section, we proposed the minimum-time scheduling problem for the cloud access nodes in the super dense wireless cloud networks. Packets originated from end users are queues in cloud access nodes to be served through the super dense wireless cloud network that provides the facility of multiple simultaneous transmissions. We modelled the cloud access node scheduling as a minimum-time scheduling problem for emptying the queued packets. We evaluated the effect of internal processing gain and traffic heterogeneity on the scheduler, and devise an ad-hoc distributed offloading mechanism to enhance the cloud utilisation factor. Numerical results prove that the proposed scheduling joint with distributed offloading mechanism ties up users QoS at a high level compared to the baseline TDMA scheme.

4.3.6 Proof of theorem

In order to solve (4.10), we show that the cost function D_i is convex in $\mathbf{r}_i$ and the set of feasible solutions defined by the constraints (4.11a) and (4.11b,c) is convex. It can be shown that $\frac{\partial D_i}{\partial r_{ji}} \geq 0$ and $\frac{\partial^2 D_i}{\partial r_{ji}^2} \geq 0$. Therefore, the Hessian matrix of D_i is positive which means that D_i is a convex function of $\mathbf{r}_i$. The Lagrangian in a compact notation is

$$L(\mathbf{r}_i, \alpha, \boldsymbol{\eta}) = \sum_{j=1}^{N} \frac{r_{ji}}{\rho_j - r_{ji}\lambda_i} - \gamma(\mathbf{r}_i{}^T \mathbf{1} - 1) - \boldsymbol{\eta}^T \mathbf{r}_i. \tag{4.15}$$

where γ and $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_N]$ are Lagrange multipliers. Regarding the KKT condition r_{ji} is optimal solution for the i-th CA nodes iff

$$\frac{\partial L}{\partial r_{ji}} = \frac{\rho_j}{(\rho_j - r_{ji}\lambda_i)^2} - \gamma - \eta_j = 0\ , \quad j = 1, \ldots, N\ , \tag{4.16}$$

$$\frac{\partial L}{\partial \gamma} = \sum_{j=1}^{N} r_{ji} = 1\ , \tag{4.17}$$

$$\eta_j r_{ji} = 0, \quad \eta_j \geq 0 \quad r_{ji} \geq 0, \quad j = 1, \ldots, N\ . \tag{4.18}$$

Equivalently,

$$\gamma = \frac{\rho_j}{(\rho_j - r_{ji}\lambda_i)^2}, \ \text{if } r_{ji} > 0\ , j = 1, \ldots, N\ , \tag{4.19}$$

$$\gamma \leq \frac{\rho_j}{(\rho_j - r_{ji}\lambda_i)^2}, \ \text{if } r_{ji} = 0\ , j = 1, \ldots, N\ . \tag{4.20}$$

If we sort the CA nodes based on their available capacity so that $(\rho_1 \geq \rho_2 \geq \rho_3)$ then there might be some CA nodes that offloading backlogged packets to them is not beneficial due to their heavy load or communication delay in the cloud backhaul. Therefore, there might be and index I so that $r_{ji} = 0$ for $j \geq I$. Regarding this fact and summing over CA nodes we have

$$\sqrt{\gamma} = \frac{\sum_{j=1}^{I-1} \sqrt{\rho_j}}{\sum_{j=1}^{I-1} \rho_j - \sum_{j=1}^{I-1} r_{ji}\lambda_i} \leq \frac{1}{\sqrt{\rho_I}}\ , \tag{4.21}$$

where I is the minimum index depicting the I-th CA node so that

$$\sqrt{\rho_I} \leq \frac{1}{\sqrt{\gamma}}\ . \tag{4.22}$$

Finally, by having the index I for each CA node we solve (4.10) for r_{ji} that is

$$r_{ji} = \begin{cases} \frac{1}{\lambda_i}\left(\rho_j - \frac{\sqrt{\rho_j}}{\sqrt{\gamma}}\right) & \text{if } 1 \leq j < I \\ 0 & \text{if } I \leq j \end{cases}\ . \tag{4.23}$$

5 Optimised transmission techniques

5.1 n-largest eigenmode relaying

5.1.1 Introduction

It was proved by van der Meulen in [93] that the capacity of a three node communication system can, potentially, be larger than the capacity of a point-to-point communication system. Consequently, analysis of communication systems in which the transceiver nodes cooperatively transmit their data to an intended final receiver has been a rather active field of research in last decade and numerous papers including, e.g. [94–97] have investigated cooperative communication systems. During the infancy of the concept of cooperative communications, substantial work was carried out, investigating cooperating nodes with single antennas. Several promising relaying protocols were proposed; among them, Amplify and Forward (AF) is intensively studied in the literature; hence, in this part, we will focus on AF relaying, too.

Employing multiple antennas in communication nodes is another technique proved to be capable of enhancing transmission rates, see [98]. Employing multiple antennas in the nodes of a cooperative communication system has been an active research trend during the last few years. Assuming multiple antennas at the relay, one major task is to design a suitable amplification matrix in the relay. Indeed, depending on the available Channel State Information (CSI) at the relay, the amplification matrix can, potentially, be different. Moreover, different communication systems can demand the optimisation of different desired performance measure; hence, different “optimal” relaying protocols will exist: for instance, the non-regenerative relaying matrices, e.g. in [99–102], are designed to minimise Mean Square Error (MSE) but other relaying matrices, e.g. in [103–107], are assumed to maximise the achievable rates.

Assuming statistical CSI at a transceiver node is interesting from a practical point of view; in particular, in rapidly changing channels, assuming perfect CSI in a relay node is, indeed, unrealistic, hence, a large body of the literature investigates AF cooperative systems wherein a single antenna relay node has access only to the statistical CSI (see [108]). Note that single antenna AF relaying systems, with statistical CSI at the relay, are usually referred to as “fixed gain" AF relaying. In spite of the importance of cooperative communication systems with statistical CSI knowledge, very few papers consider the problem when the relay node is equipped with multiple antennas. Moreover, except [103], we are not aware of any other paper assuming fading correlation in the relay when only the covariance of the channels is known to the relay. Note that fading correlation at a transceiver can be due to an unobstructed node or space limits at the node which forces the antennas to be closely located. Justifications to assume transceivers with fading correlation can be found in [109, 110].

The first contribution of this part is to provide a statistical analysis of the received Signal-to-Noise Ratio (SNR) at the destination. There are two major motivations for studying the statistical characteristics of the SNR:

- Outage probability is directly related to received SNR. Indeed, the cumulative density function (cdf) of the SNR corresponds to the outage probability, and so, the cdf of SNR will be derived in this part.

- By deriving the cdf of SNR, the mathematical complexity of direct maximisation of the achievable rate, i.e. “optimal” power allocation, will be revealed. It will be an excellent motivation for devising alternative approaches with reasonable complexity.

Accordingly, the second major contribution of this part is to study the problem of power allocation in the relay, and hence to devise a new and simple power allocation scheme for multi-antenna relays. [103, 106, 107, 111] consider the similar problem of the power allocation in the relay when statistical CSI is available in the nodes (either the source or relay nodes). However, while [107] considers the high SNR regime of the system, [106, 111] assumes correlation at the source node. In [103], we study a cooperative communication system wherein the relay node is equipped with multiple antennas that are spatially correlated. The considered system is studied only at low SNR and it is proved that Largest Eigenmode Relaying (LER) is the optimal transmission method at low SNR; however, the system is not studied in the moderate and high SNR region. To the best of our knowledge, the design of an amplification matrix in an AF cooperative system where *only* the statistical CSI is known to the relay is an open problem, and one that will be tackled in this part. We provide a scheme which operates in the regime beyond that where LER is optimal, and whose performance is indistinguishable from the benchmark provided by exhaustive search.

This part is organised as follows: In Section 5.1.2, the system model is introduced and some preliminary existing results are recalled. Section 5.1.4 deals with characterising the statistics of the SNR at the destination. In Section 5.1.5, a simple power allocation algorithm is introduced for a relay with only two antennas; the proposed algorithm is called “proportional power allocation" and has been extended for a system with multiple antenna relay node in Section 5.1.6 and 5.1.7 and, finally, the results are summarised in Section 5.1.8.

5.1.2 System model and preliminaries

5.1.3 Notation

Matrices are represented by boldface upper cases ($\boldsymbol{H}$). Column and row vectors are denoted by boldface lower cases ($\boldsymbol{h}$), and h_i indicates the i-th element of $\boldsymbol{h}$. The superscript $(\,\cdot\,)^H$ stands for Hermitian transposition. We refer to the identity matrix by $\boldsymbol{I}$. The expectation operation is indicated by $\mathbb{E}\{\,\cdot\,\}$, the probability of a random variable is indicated by $\mathbb{P}(\,\cdot\,)$ and $f_X(x)$ is reserved for probability density functions (pdf) of random variable X; $\boldsymbol{\Lambda}_\Sigma$ represents a diagonal matrix with elements organised in descending order and λ_i^Σ denotes the i-th diagonal element of $\boldsymbol{\Lambda}_\Sigma$. For simplicity of notation, $(\lambda_i^\Sigma)^2$ is abbreviated by $\lambda_i^{\Sigma 2}$. The trace of a matrix is denoted by $\mathrm{Tr}(\,\cdot\,)$.

System model

In this part a dual hop, half duplex MIMO communication system is investigated. Assume a source node (equipped with n_S antennas) transmits data to a single antenna destination via an intermediate relay node which has n_R antennas. The proposed system models the downlink of a wide range of communication systems in which the user terminal is equipped with single antenna due to space limitation, for instance, cellular networks or sensor networks. Moreover, *fixed-gain* AF cooperative systems with multiple antennas at the relay is an open problem which has received little attention and so the proposed system model is a good step forward for understanding fixed gain AF systems. It is assumed that a direct link between the source and the destination is not available. The half duplex constraint is accomplished by time sharing between the source and the relay; i.e. each transmission period is divided into two time slots: the source transmits during the first time slot and the relay during the second one. The relay remains silent during the source transmission and vice versa. It is assumed that the source does not have access to any statistical or instantaneous channel state information (CSI). Moreover, it is assumed that the antennas in the source node are sufficiently far apart and so no correlation is assumed at the source. The signal received at the relay ($\boldsymbol{y}_\mathrm{R}$) due to the source transmission is given by

$$\boldsymbol{y}_\mathrm{R} = \boldsymbol{H}_1 \boldsymbol{x} + \boldsymbol{w}_\mathrm{R} \tag{5.1}$$

where the $n_\mathrm{R} \times n_\mathrm{S}$ matrix $\boldsymbol{H}_1$ represents the channel between the source and the relay. With P_S the power constraint of the source, the column vector $\boldsymbol{x}$ is the signal transmitted from the source with $\boldsymbol{Q} = \mathbb{E}(\boldsymbol{x}\boldsymbol{x}^H) = \frac{P_\mathrm{S}}{n_\mathrm{S}} \boldsymbol{I}_{n_\mathrm{S}}$ and the column vector $\boldsymbol{w}_\mathrm{R}$ represents the receiver noise in the relay with elements independently drawn from a complex Gaussian random variable with variance N_0. In this part, it is assumed that spatial correlation occurs at the relay; the correlation can be due to space limit in the relay or due to fading correlation due to unobstructed relay node. $\mathbf{\Sigma}$ represents the correlation matrix at the relay and therefore, using the Kronecker model, $\boldsymbol{H}_1$ can be written as

$$\boldsymbol{H}_1 = \mathbf{\Sigma}^{\frac{1}{2}} \boldsymbol{H}_{1w} \tag{5.2}$$

where elements of $\boldsymbol{H}_{1w}$ are i.i.d., zero mean, unit variance complex Gaussian random variables, independent of each other. The relay multiplies $\boldsymbol{y}_{\mathrm{R}}$ by the gain matrix $\boldsymbol{F}$ and forwards it to the destination. Then, the received signal at the destination is

$$\begin{aligned} y_{\mathrm{D}} &= \boldsymbol{h}_2 \boldsymbol{F} \boldsymbol{y}_{\mathrm{R}} + w_{\mathrm{D}} \\ &= \boldsymbol{h}_2 \boldsymbol{F} \boldsymbol{H}_1 \boldsymbol{x} + \boldsymbol{h}_2 \boldsymbol{F} \boldsymbol{w}_{\mathrm{R}} + w_{\mathrm{D}} \end{aligned} \tag{5.3}$$

where the row vector $\boldsymbol{h}_2$ indicates the channel between the relay and the destination; w_{D} represents the receiver noise at the destination. For simplicity, we assume that w_{D} and $\boldsymbol{w}_{\mathrm{R}}$ are statistically independent and identical, i.e. $N_{0,w_{\mathrm{D}}} = N_{0,\boldsymbol{w}_{\mathrm{R}}} = N_0$.
Due to the spatial correlation $\mathbf{\Sigma}$ at the relay, one can factorise $\boldsymbol{h}_2$ as

$$\boldsymbol{h}_2 = \boldsymbol{h}_{2w} \mathbf{\Sigma}^{\frac{1}{2}} \tag{5.4}$$

where elements of $\boldsymbol{h}_{2w}$ are i.i.d., zero mean, unit variance complex Gaussian random variables, independent of each other. Justification to assume transceivers with spatial correlation can be found in [109, 110]. The correlation matrix $\mathbf{\Sigma}$ in the relay is decomposed using spectral decomposition as

$$\mathbf{\Sigma} = \boldsymbol{U}_{\Sigma} \mathbf{\Lambda}_{\Sigma} \boldsymbol{U}_{\Sigma}^{H} \tag{5.5}$$

where $\boldsymbol{U}_{\Sigma}$ is a unitary matrix whose columns are the eigenvectors corresponding to $\mathbf{\Sigma}$, and $\mathbf{\Lambda}_{\Sigma}$ is a diagonal matrix with the eigenvalues of $\mathbf{\Sigma}$ in decreasing order, i.e.

$$\mathbf{\Lambda}_{\Sigma} = \mathrm{diag}[\lambda_1^{\Sigma}, \lambda_2^{\Sigma}, \cdots, \lambda_{n_{\mathrm{R}}}^{\Sigma}],$$

where $\lambda_1^{\Sigma} \geq \lambda_2^{\Sigma} \geq \cdots \geq \lambda_{n_{\mathrm{R}}}^{\Sigma} \geq 0$. Moreover, some of λ_i^{Σ}s *can* possibly be zero.

Preliminaries

Ergodic capacity is one of the main performance criterion investigated in this part. Using (5.3), the ergodic capacity of the system is defined as

$$C_{av} = \frac{1}{2} \max_{\substack{\boldsymbol{Q} = \frac{P_{\mathrm{S}}}{n_{\mathrm{S}}} \boldsymbol{I} \\ F : \mathbb{E}\{\|\boldsymbol{F}\boldsymbol{y}_{\mathrm{R}}\|^2\} \leq P_{\mathrm{R}}}} \mathbb{E}\{C(\boldsymbol{H}_1, \boldsymbol{h}_2, \boldsymbol{F})\} \tag{5.6}$$

where $C(\boldsymbol{H}_1, \boldsymbol{h}_2, \boldsymbol{F})$ is the conditional transmission rate. For simplicity of notation, $C(\boldsymbol{H}_1, \boldsymbol{h}_2, \boldsymbol{F})$ is abbreviated by $C(\,\cdot\,)$ in the rest of this part. Assuming perfect CSI of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ at the destination and $\boldsymbol{Q} = \frac{P_{\mathrm{S}}}{n_{\mathrm{S}}} \boldsymbol{I}_{n_{\mathrm{S}}}$ (equal transmit power from each antenna in the source, because no channel knowledge is available there), the conditional mutual information $C(\,\cdot\,)$ for given channel matrices is

$$C(\,\cdot\,) = \log\big(1 + \frac{P_{\mathrm{S}}}{n_{\mathrm{S}}}\frac{\boldsymbol{h}_2\boldsymbol{F}\boldsymbol{H}_1\boldsymbol{H}_1^H\boldsymbol{F}^H\boldsymbol{h}_2^H}{N_0(1+\boldsymbol{h}_2\boldsymbol{F}\boldsymbol{F}^H\boldsymbol{h}_2^H)}\big) \tag{5.7}$$

where $N_0(1+\boldsymbol{h}_2\boldsymbol{F}\boldsymbol{F}^H\boldsymbol{h}_2^H)$ is the total equivalent noise power which is assumed to remain constant for coherence time: we make a block fading assumption. In [103], $\boldsymbol{F}$ is found to be symmetric as

$$\boldsymbol{F} = \boldsymbol{G}^{\frac{1}{2}} \tag{5.8}$$

where the gain matrix $\boldsymbol{G}$ is derived as

$$\boldsymbol{G} = \boldsymbol{U}_\Sigma\boldsymbol{\Lambda}_G\boldsymbol{U}_\Sigma^H \tag{5.9}$$

where $\boldsymbol{U}_\Sigma$ is the unitary matrix defined in (5.5) and $\boldsymbol{\Lambda}_G = \mathrm{diag}[\lambda_1^G, \lambda_2^G, \cdots, \lambda_{n_{\mathrm{R}}}^G]$. Note that λ_i^G values are to be specified according to the power constraint of the relay so that the maximisation in (5.6) is accomplished; indeed, this is one of the main tasks to be handled in this part.

Assuming (5.4), (5.5), (5.8) and (5.9), the power constraint in the relay i.e.

$$\mathbb{E}\{\|\boldsymbol{G}^{\frac{1}{2}}\boldsymbol{y}_{\mathrm{R}}\|^2\} \leq P_{\mathrm{R}}$$

in (5.6)) is

$$P_{\mathrm{S}}\mathrm{Tr}(\boldsymbol{\Lambda}_\Sigma\boldsymbol{\Lambda}_G) + N_0\mathrm{Tr}(\boldsymbol{\Lambda}_G) = P_{\mathrm{R}}. \tag{5.10}$$

Note that the capacity will be achieved by consuming the entire power at the relay, and so, we assume –equality– in (5.10) instead of –inequality–. By combining (5.2), (5.4), (5.7), (5.8) and assuming $\gamma_{\mathrm{S}} = P_{\mathrm{S}}/N_0$, one can write (5.7) as follows

$$C(\,\cdot\,) = \log(1 + \underbrace{\frac{\gamma_{\mathrm{S}}\boldsymbol{h}_{2w}\boldsymbol{\Lambda}_\Sigma\boldsymbol{\Lambda}_G^{\frac{1}{2}}\boldsymbol{H}_{1w}\boldsymbol{H}_{1w}^H\boldsymbol{\Lambda}_\Sigma\boldsymbol{\Lambda}_G^{\frac{1}{2}}\boldsymbol{h}_{2w}^H}{n_{\mathrm{S}}(1+\boldsymbol{h}_{2w}\boldsymbol{\Lambda}_\Sigma\boldsymbol{\Lambda}_G\boldsymbol{h}_{2w}^H)}}_{\gamma_{\mathrm{D}}}) \tag{5.11}$$

where γ_{D} represents received SNR in the destination which is a function of $\boldsymbol{H}_{1w}$, $\boldsymbol{h}_{2w}$, the correlation eigenvalues matrix $\boldsymbol{\Lambda}_\Sigma$ and the eigenvalues of the $\boldsymbol{G}$ matrix, i.e. $\boldsymbol{\Lambda}_G$. γ_{D} can be simplified according to

$$\gamma_{\mathrm{D}} = \frac{\gamma_{\mathrm{S}}\sum_{i=1}^{n_{\mathrm{S}}} \mid \boldsymbol{h}_{2w}\boldsymbol{\Lambda}_G^{\frac{1}{2}}\boldsymbol{\Lambda}_\Sigma\boldsymbol{h}_{1w,i} \mid^2}{n_{\mathrm{S}}(1+\boldsymbol{h}_{2w}\boldsymbol{\Lambda}_G\boldsymbol{\Lambda}_\Sigma\boldsymbol{h}_{2w}^H)} \tag{5.12}$$

where $\boldsymbol{h}_{1w,i}$ represents the ith column of $\boldsymbol{H}_{1w}$. Let us assume

$$X_j = |h_{2w,j}|^2 \tag{5.13a}$$

$$Y = \frac{1}{n_\mathrm{S}} \sum_{i=1}^{n_\mathrm{S}} |h_{1w,i}|^2 \tag{5.13b}$$

It is proved in [103, Appendix 1] that γ_D in (5.12) can be further simplified to

$$\gamma_\mathrm{D} = \gamma_\mathrm{S} Y \times \underbrace{\frac{\sum\limits_{j=1}^{\kappa} \lambda_j^G \lambda_j^{\Sigma 2} X_j}{1 + \sum\limits_{j=1}^{\kappa} \lambda_j^G \lambda_j^{\Sigma} X_j}}_{X} \tag{5.14}$$

where κ is the minimum of n_R and Number of Non-Zero (nnz) λ_j^Σ, i.e.,

$$\kappa = \min(n_\mathrm{R}, \mathrm{nnz}(\lambda_j^\Sigma)) \tag{5.15}$$

Note that Y and X_j correspond to the S-R and R-D channels, respectively. The random variable

$$X = \frac{\sum\limits_{j=1}^{\kappa} \lambda_j^G \lambda_j^{\Sigma 2} X_j}{1 + \sum\limits_{j=1}^{\kappa} \lambda_j^G \lambda_j^{\Sigma} X_j} \tag{5.16}$$

in (5.14) incorporates the effect of the R-D link as well as the effect of power allocation due to λ_j^G. Furthermore, since we assume Rayleigh fading in both the S-R and R-D links, hence, X_j is exponentially distributed with unit mean and Y has an Erlang-distribution with rate and shape parameters equal to n_S.

Although it is proposed in [103] that the optimal $\boldsymbol{G}$ should be diagonalised according to (5.9) where $\boldsymbol{\Lambda}_G$ is a diagonal matrix with its components organised in descending order, an optimal power allocation method to distribute relay's transmit power among different λ_j^Gs is not discussed. That is still an open problem but will be addressed in this part.

Contribution

There are two main problems investigated in the following sections, each leading to novel contributions:

- We evaluate, for the first time, the statistical characteristics of the γ_D introduced in (5.14). Due to its mathematical complexity, the exact pdf of γ_D is not derived but an approximation to the pdf is provided in this work. The approximated pdf is then

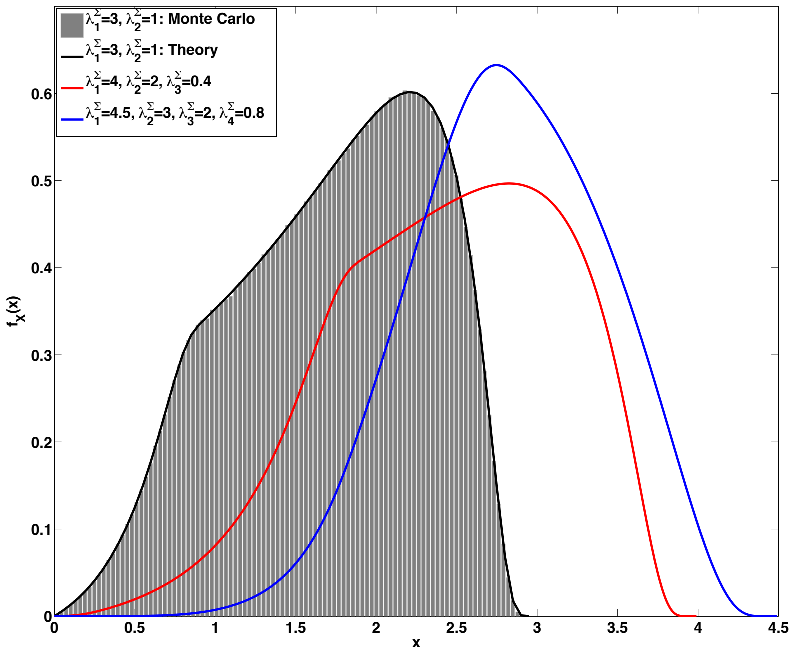


Figure 5.1: Pdf of X with various number of λ_j^Σ parameters. Monte Carlo simulations validate the correctness of the theoretical results.

used for calculating the outage probability and it is illustrated, by the simulations, that the approximated cdf leads to rather accurate results. Moreover, the exact cdf of γ_{D} will be derived for the two asymptotic scenarios of full-correlation and no-correlation at the relay. Although this novel cdf is helpful for outage analysis of the system, it is too complicated to be used for the analysis of the ergodic capacity.

- In order to approximate the maximum achievable transmission rate, a very simple power allocation algorithm at the relay is introduced in this part. As the optimal power allocation at moderate and high SNR is still an open problem[1], for the purpose of comparison, exhaustive search over various discrete values of the rates is used as a benchmark. The values of the rates are obtained by allocating various amount of the power among different eigenmodes. According to the simulations, the proposed power allocation algorithm approximates the benchmark with insignificant difference.

5.1.4 Statistical analysis of received SNR at destination

The SNR distribution in the relay is directly related to the ergodic capacity of the system evaluated in this part. In order to maximise $\mathbb{E}\left\{C(\,\cdot\,)\right\}$ in (5.6), one should distribute the

[1] The optimal power allocation at low SNR was proposed in [103].

available relay power appropriately among different λ_j^Gs in (5.14) so that $\mathbb{E}\{C(\,\cdot\,)\}$ is maximised. Before we continue with the statistical characterisation of γ_D, two extreme scenarios are studied: we will investigate γ_D when the relay does not experience any fading correlation and also when the antennas in the relay are fully correlated. These two scenarios, indeed, provide performance bounds, and so, the performance of a system with partial correlation will fall between the two bounds.

Statistical characteristics of γ_D assuming full correlation

Full correlation (FC) at the relay is equivalent to considering a system where all the elements of the $\mathbf{\Sigma}$ are unity[2], i.e. $\mathbf{\Sigma} = \mathbf{1}_{n_\mathrm{R} \times n_\mathrm{R}}$. Consequently, it is easy to see that $\lambda_1^\Sigma = n_\mathrm{R}$ and $\lambda_j^\Sigma = 0$ for $j \geq 2$, and so, the random variable X in (5.16) can be written as

$$X_{FC} = \frac{\lambda_1^G \lambda_1^{\Sigma 2} X_1}{1 + \lambda_1^G \lambda_1^\Sigma X_1} = \frac{n_\mathrm{R}^2 \lambda_1^G X_1}{1 + n_\mathrm{R} \lambda_1^G X_1} = n_\mathrm{R} \frac{V}{1+V} \tag{5.17}$$

where, the random variable $V = n_\mathrm{R} \lambda_1^G X_1$ is exponentially distributed with mean $n_\mathrm{R} \lambda_1^G$. The following theorem introduces the cdf of γ_D for a fully correlated relay:

Theorem 5.1. *Assuming fully correlated antennas at the relay, the cdf of γ_D is*

$$F_{\gamma_{D-FC}}(x) = 1 - 2(n_S w)^{n_S} \mathrm{e}^{-n_S w} \sum_{m=0}^{n_S - 1} \frac{(\lambda_1^G n_S n_R w)^{-(m+1)/2}}{m!(n_S - m - 1)!} \mathrm{K}_{m+1}(2\sqrt{\frac{n_S w}{\lambda_1^G n_R}})$$

where $\mathrm{K}_\nu(\,\cdot\,)$ is the modified Bessel function of the second kind, ν-th order and $w = \frac{x}{n_\mathrm{R} \gamma_\mathrm{S}}$.

Proof. See section 5.1.9 for a detailed proof. □

The subscript “FC" in $F_{\gamma_{D-FC}}(x)$ indicates the Full Correlation scenario. One can easily derive the pdf of γ_D for the full correlation scenario by taking the derivative of (5.18) with respect to x.

In the next section, we study the statistical characteristics of the system for the non-correlated fading scenario.

[2] We assume normalised correlation for simplicity of the notation. Generalisation to a case with non-unity full correlation is straightforward. Similar normalisation is assumed in Section 5.1.4, too.

Statistical characteristics of γ_{D} assuming no correlation

No correlation (NC) at the relay translates to $\boldsymbol{\Sigma} = \boldsymbol{I}$. Such a scenario will occur when the relay antennas are placed sufficiently far apart and that the relay node is placed in a rich scattering environment. Assuming $\boldsymbol{\Sigma} = \boldsymbol{I}$, it is straightforward to conclude that $\lambda_1^{\Sigma} = \lambda_1^{\Sigma} = \cdots = \lambda_{n_{\mathrm{R}}}^{\Sigma} = 1$. On the other hand, since all X_j random variables follow the same distribution (exponential distribution with unit mean) and as all λ_j^{Σ} values are equal to one, therefore all the λ_j^G values should be assigned the same power, and so, let us assume $\lambda_1^G = \lambda_2^G = \cdots = \lambda_{n_{\mathrm{R}}}^G = \lambda_{\mathrm{eq}}^G$; consequently, the random variable X will be written as

$$X_{NC} = \frac{\lambda_{\mathrm{eq}}^G \sum_{j=1}^{n_{\mathrm{R}}} X_j}{1 + \lambda_{\mathrm{eq}}^G \sum_{j=1}^{n_{\mathrm{R}}} X_j} = \frac{V}{1+V} \tag{5.18}$$

where V follows the Erlang distribution with rate $\frac{1}{\lambda_{\mathrm{eq}}^G}$ and shape n_{R}. By substituting X_{NC} in (5.14), the following expression will be derived for the cdf of γ_{D} for the non correlated scenario:

Theorem 5.2. *Assuming no correlation at the relay, the cdf of γ_D is*

$$F_{\gamma_{D-NC}}(x) \quad = \quad 1 - 2(n_S w)^{n_S} \mathrm{e}^{-n_S w} \sum_{m=0}^{n_R-1} \sum_{n=0}^{n_S-1} \frac{n_S^m (\lambda_{eq}^G n_S w)^{-\frac{m+n+1}{2}}}{m!n!(n_S-n-1)!w^m} \mathrm{K}_{m-n-1}(2\sqrt{\frac{n_S w}{\lambda_{eq}^G}})$$

with $w = \frac{x}{\gamma_{\mathrm{S}}}$.

Proof. Following the same lines of the proof for (5.18), one can easily prove (5.19), too. □

Assuming single antennas at the source and the relay nodes, i.e. $n_{\mathrm{S}} = n_{\mathrm{R}} = 1$, the MIMO scenario of this part reduces to a conventional single antenna relay system wherein the relay node has access to the variance of its channels; consequently, as expected, the two expressions in (5.18) and (5.19) are identical according to

$$F_{\gamma_{\mathrm{D}}}(x) = 1 - 2\sqrt{\frac{(1+\gamma_{\mathrm{S}})x}{\gamma_{\mathrm{S}}\gamma_{\mathrm{R}}}} \mathrm{e}^{-\frac{x}{\gamma_{\mathrm{S}}}} \mathrm{K}_1(2\sqrt{\frac{(1+\gamma_{\mathrm{S}})x}{\gamma_{\mathrm{S}}\gamma_{\mathrm{R}}}}) \tag{5.19}$$

Note that $F_{\gamma_{\mathrm{D}}}(x)$ for a single antenna scenario similar to (5.19) has been reported in numerous parts including, e.g. [112].

As discussed earlier, $F_{\gamma_{\mathrm{D}}}$ (and $f_{\gamma_{\mathrm{D}}}$) for arbitrary correlation should follow an expression that at the extreme case reduces to (5.19) and (5.18).

On the other hand, statistical characteristics of X are also of essential importance for understanding the distribution γ_{D}. Assuming full correlation and no correlation at the relay, characterising X was simple, however, characterising X with arbitrary correlation is, actually, more complicated and will be derived in the next section.

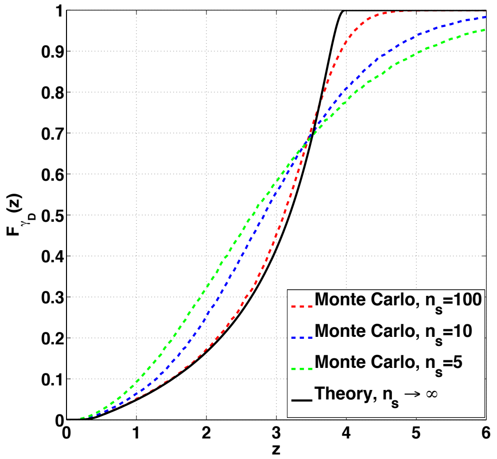


Figure 5.2: Approximation of $F_{\gamma_D}(z)$ as obtained in (5.26) where $\lambda_1^{\Sigma} = 4$, $\lambda_2^{\Sigma} = 1$ and $\lambda_1^G = \lambda_2^G = \lambda_{\text{eq}}^G = 2$.

Statistical characteristics of X for arbitrary correlation

The statistical characteristics of the random variable X are not studied in the literature but will be derived in this part.

Considering that $\lambda_1^{\Sigma} \geq \lambda_2^{\Sigma} \geq \cdots \geq \lambda_{\kappa}^{\Sigma} \geq 0$, the cdf of X in (5.16) is

$$F_X(x) = \begin{cases} 0 & x \leq 0 \\ 1 - \sum\limits_{i=1}^{j} \prod\limits_{\substack{m=1 \\ m \neq j}}^{\kappa} \frac{c_m}{c_m - c_j} \mathrm{e}^{c_i x} & \lambda_{j+1}^{\Sigma} < x < \lambda_j^{\Sigma} \\ 1 & \lambda_1^{\Sigma} \leq x \end{cases} \tag{5.20}$$

where

$$c_j = \frac{1}{\lambda_j^G \lambda_j^{\Sigma} (\lambda_j^{\Sigma} - x)}. \tag{5.21}$$

A sketch of the proof for (5.20) is provided in Section 5.1.10. To provide a better understanding of the statistical characteristics of the random variable X, the $F_X(x)$ for $\kappa = 2$ and 3 is provided in the following. Assuming $\kappa = 2$, we have

$$F_X(x) = \begin{cases} 0 & x \leq 0 \\ 1 - \frac{c_2}{c_2 - c_1} \mathrm{e}^{c_1 x} - \frac{c_1}{c_1 - c_2} \mathrm{e}^{c_2 x} & 0 < x < \lambda_2^{\Sigma} \\ 1 - \frac{c_2}{c_2 - c_1} \mathrm{e}^{c_1 x} & \lambda_2^{\Sigma} < x < \lambda_1^{\Sigma} \\ 1 & x \geq \lambda_1^{\Sigma} \end{cases} \tag{5.22}$$

and for $\kappa = 3$, $F_X(x)$ is derived in (5.23) at the top of next page. One can easily derive

$$F_X(x) = \begin{cases} 0 & x \leq 0 \\ 1 - \frac{c_2 c_3}{(c_2 - c_1)(c_3 - c_1)} \mathrm{e}^{c_1 x} - \frac{c_1 c_3}{(c_3 - c_2)(c_1 - c_2)} \mathrm{e}^{c_2 x} - \frac{c_1 c_2}{(c_1 - c_3)(c_2 - c_3)} \mathrm{e}^{c_3 x} & 0 < x < \lambda_3^{\Sigma} \\ 1 - \frac{c_2 c_3}{(c_2 - c_1)(c_3 - c_1)} \mathrm{e}^{c_1 x} - \frac{c_1 c_3}{(c_3 - c_2)(c_1 - c_2)} \mathrm{e}^{c_2 x} & \lambda_3^{\Sigma} < x < \lambda_2^{\Sigma} \\ 1 - \frac{c_2 c_3}{(c_2 - c_1)(c_3 - c_1)} \mathrm{e}^{c_1 x} & \lambda_2^{\Sigma} < x < \lambda_1^{\Sigma} \\ 1 & x \geq \lambda_1^{\Sigma} \end{cases} \tag{5.23}$$

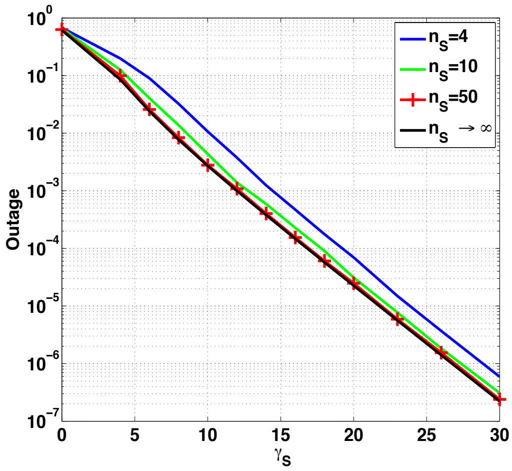


Figure 5.3: Outage probability approximated by (5.26) ($n_\mathrm{S} \to \infty$) in comparison with the Monte Carlo simulations for various values of n_S.

$f_X(x)$ by taking the derivative of $F_X(x)$ with respect to x, i.e.,

$$f_X(x) = \frac{\mathrm{d}F_X(x)}{\mathrm{d}x} \tag{5.24}$$

which, is a straightforward simple derivation practice. Figure 5.1 illustrates the $f_X(x)$ assuming $\lambda_j^G = 1$ for various values of λ_j^Σ. The agreement between the theoretical and Monte Carlo simulations validates the correctness of the calculations.

Statistical characteristics of γ_D assuming infinite antennas at the source

From the discussions provided in the previous section, it is clear that $\gamma_\mathrm{D} = \gamma_\mathrm{S} Y X$ where Y follows an Erlang distribution with rate and shape parameters equal to n_S, i.e.

$$f_Y(y) = \frac{n_\mathrm{S}^{n_\mathrm{S}}}{(n_\mathrm{S} - 1)!} y^{n_\mathrm{S} - 1} \mathrm{e}^{-n_\mathrm{S} y}$$

and X follows a distribution as derived in (5.20) and (5.24). In order to calculate the cdf of γ_{D}, one should calculate the following integral

$$\begin{aligned} F_{\gamma_{\mathrm{D}}}(z) &= \mathbb{P}\{\gamma_{\mathrm{D}} < z\} = \mathbb{P}\{\gamma_{\mathrm{S}} Y X < z\} \\ &= \int_0^{\infty} \underbrace{\mathbb{P}\{X < \frac{z}{y\gamma_{\mathrm{S}}}\}}_{F_X(\frac{z}{y\gamma_{\mathrm{S}}})} f_Y(y)\mathrm{d}y. \end{aligned} \tag{5.25}$$

On the other hand, (5.25) does not lend itself easily to further calculations, and to the best of our knowledge, the integral cannot be solved with the existing table of integrals, e.g. [113]. However, clearly, $F_{\gamma_{\mathrm{D}}}(z)$ is a multipartite function because X has a multipartite cdf; to the best of our knowledge, a multipartite $F_{\gamma_{\mathrm{D}}}(z)$ in the context of AF cooperative systems has not been reported in the literature and so it is observed for the first time in this part[3]. The following theorem provides an approximate $F_{\gamma_{\mathrm{D}}}(z)$ for the given system model:

Theorem 5.3. *For large n_S, $F_{\gamma_D}(z)$ can be approximated by*

$$F_{\gamma_D}(z) \approx F_X(\frac{z}{\gamma_S}) \tag{5.26}$$

Proof. The random variable Y in the previous sections, e.g. in (5.14), follows an Erlang-distribution; indeed, Y is the sum of n_{S} exponential random variables, each with parameter n_{S} (see (5.13a)). Using the central limit theorem (see [114, Ch. 7.4]), the random variable Y can be approximated by a Gaussian distribution with mean equal to 1 and variance $1/n_{\mathrm{S}}$, i.e. approximately, $Y \sim \mathcal{N}(1, 1/n_{\mathrm{S}})$. Assuming large n_{S}, i.e. $n_{\mathrm{S}} \to \infty$, one can easily deduce $Y \to 1$. By setting $Y = 1$ in (5.25), one can write

$$F_{\gamma_{\mathrm{D}}}(z) = \mathbb{P}\{\gamma_{\mathrm{D}} < z\} = \mathbb{P}\{X < \frac{z}{\gamma_{\mathrm{S}}}\}$$

and so (5.26) is proved. □

Figure 5.2 and Figure 5.3 are intended to validate the precision of the approximation obtained in (5.26). In Figure 5.2 an illustration of $F_{\gamma_D}(z)$ is provided; clearly, Monte Carlo simulations approximate theoretical $F_{\gamma_D}(z)$ when n_{S} is large. Moreover, considering that the cdf $F_{\gamma_{\mathrm{D}}(z)}$ in (5.26) corresponds to the outage probability for large n_{S} (ideally for $n_{\mathrm{S}} \to \infty$), in Figure 5.3 we plot outage probability versus transmit power at the source node using (5.26) and also using Monte Carlo simulations for various values of n_{S}. It is clear that for large values of n_{S}, the closed form expression for the outage probability

[3] We use multipartite function to refer to a function that involves several distinct functions for different domains, e.g. see (5.22) and (5.23). Note that being multipartite is not considered to be advantage (or disadvantage) for the system but stressing on the novelty of the statistical characteristics of the SNR, in the context of AF cooperative systems, is meant to highlight the need for further investigation on the problem.

approximates the Monte Carlo simulations with high accuracy. Nevertheless, for smaller values of the n_{S}, although the approximation is not accurate, it provides a reasonable approximation.

As mentioned in 5.1.3, one of the main objectives in this part is to allocate available power in the relay according to (5.10) among different λ_j^G variables so that the ergodic capacity in (5.6) is maximised. In fact, for plotting Figure 5.1 we arbitrarily assumed $\lambda_1^G = \lambda_2^G = 1$; however, such a random power allocation to λ_1^G and λ_2^G does not guarantee that the maximisation problem in (5.6) is solved. On the other hand, it was observed in (5.25) that the $F_X(\,\cdot\,)$ expression derived in (5.20) is too complicated to lend itself to further mathematical calculations. Indeed, we do not know any closed form expression for the objective function $\mathbb{E}\{C(\,\cdot\,)\}$ which can be used for calculating optimal λ_j^G values in (5.6). Furthermore, not only do we not know any analytical way for calculating optimal λ_j^G values, we are not aware of any numerical method to calculate optimal λ_j^G values. For the purpose of comparison, exhaustive search over various discrete values of the achievable rates is used as a benchmark. The rates for the exhaustive search are obtained by assigning various amount of the power to the eigenmodes according to the power constraint in (5.10); moreover, the resolution of the exhaustive search is kept adequately small (0.1 dB) to ensure accurate approximation. Note that resolutions larger than 0.1 dB also provide accurate results, however, to make sure that no local maximum is missed, we use the the resolution of 0.1 dB throughout this part when exhaustive search is provided for comparison.

Although the proposed method in the next section is simple and straightforward, it will be revealed that the obtained values for λ_j^Gs lead to the reasonable rates that are indistinguishable from the benchmark rates. Also, it will be revealed that the proposed method significantly reduces the computationally expensive calculations due to exhaustive search for finding optimal λ_j^G values in real time practical communication systems.

5.1.5 Two antenna relay

For simplicity, as an initial step, let us assume a system with two antennas at the relay, i.e. $n_{\mathrm{R}} = 2$, where $\mathbf{\Sigma} = [\begin{smallmatrix} 1 & \rho \\ \rho^* & 1 \end{smallmatrix}]$; the parameter ρ indicates the correlation coefficient. As there are only two eigenvectors corresponding to $\mathbf{\Sigma}$, the problem of optimal power allocation reduces to calculating the optimal values of λ_1^G and λ_2^G, given the power constraint in (5.10).

In [103, Eq. 34], we derive a necessary and sufficient condition under which transmission *only* from the largest eigenvector (the eigenvector corresponding to λ_1^Σ) achieves capacity. For ease of reference, [103, Eq. 34] is provided in the following lemma:

Lemma: *Transmission from the largest eigenvector achieves capacity if*

$$\lambda_2^\Sigma \leq \frac{(\alpha_1 + P_1\lambda_2^\Sigma)\mathcal{D}(\lambda_1^\Sigma, P_1) - \alpha_1\mathbb{E}\{\frac{1}{1+P_1Z_1}\}}{P_1\mathbb{E}\{\frac{1+\lambda_2^\Sigma\gamma Y}{1+P_1Z_1}\}} \tag{5.27}$$

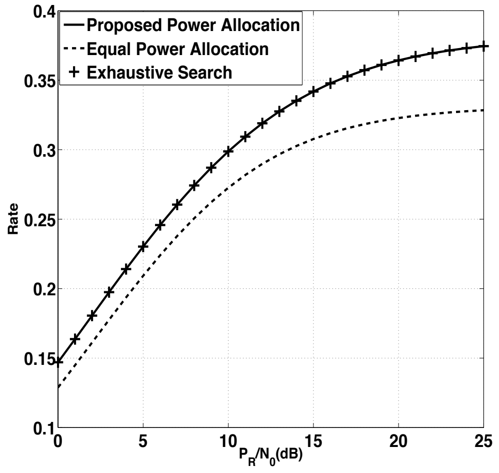


Figure 5.4: Rate vs. transmit power at the relay. The rate values are obtained using the proposed power allocation algorithm, equal power allocation and the benchmark. $n_S = n_R = 2$, $N_0 = 1$, $P_S = 0\,\text{dB}$, and inter-antenna correlation $\rho = 0.3$. The exhaustive search is performed over discrete values of transmission rates that were obtained using various values of λ_1^G and λ_2^G (step size 0.1 dB) that fulfil (5.10).

with

$$Z_1 = \lambda_1^\Sigma(1 + \gamma\lambda_1^\Sigma Y)X_1 \tag{5.28}$$

$$\mathcal{D}(\lambda_1^\Sigma, P_1) = \frac{1}{P_1\lambda_1^\Sigma}\Gamma(0, \frac{1}{P_1\lambda_1^\Sigma})\mathrm{e}^{\frac{1}{P_1\lambda_1^\Sigma}} \tag{5.29}$$

where $P_1 = \frac{P_R}{\lambda_1^\Sigma P_S + N_0}$ and $\alpha_1 = \frac{\lambda_2^\Sigma P_S + N_0}{\lambda_1^\Sigma P_S + N_0}$. Although [103] proves that LER is the optimal transmission method at low SNR, it does not discuss any method to distribute the available power at the relay among λ_j^G variables when (5.27) does not hold. *This problem is addressed in the rest of this part.*

Proposition. Given λ_1^Σ, λ_2^Σ, P_S and P_R , if (5.27) holds, allocate the entire power in the relay only to λ_1^G as

$$\lambda_1^G = \frac{P_R}{N_0 + P_S\lambda_1^\Sigma} \tag{5.30}$$

and set $\lambda_2^G = 0$, i.e. LER, otherwise, when (5.27) does *not* hold, we propose to allocate power per eigenvector proportionally to the strength of the eigenmodes, i.e.

$$\frac{\lambda_1^G}{\lambda_2^G} = \frac{\lambda_1^\Sigma}{\lambda_2^\Sigma}. \tag{5.31}$$

Consequently, one can assume $\lambda_1^G = \lambda_1^\Sigma g_2$ and $\lambda_2^G = \lambda_2^\Sigma g_2$ with g_2 obtained from (5.10) as

$$g_2 = \frac{P_\mathrm{R}}{P_\mathrm{S}(\lambda_1^{\Sigma 2} + \lambda_2^{\Sigma 2}) + N_0(\lambda_1^\Sigma + \lambda_2^\Sigma)}. \tag{5.32}$$

Note that (5.30) and (5.32) are obtained from the relay power constraint in (5.10). ■

Remark: The motivation for assuming proportional power allocation in (5.31) arises from the limit behaviour of the correlation coefficients. In the case where λ_1^Σ much larger than λ_2^Σ, i.e. $\lambda_1^\Sigma \gg \lambda_2^\Sigma$, clearly LER will be optimal transmission method and so $\lambda_1^G \gg \lambda_2^G$ has to hold, and this is guaranteed by (5.31). On the other hand when λ_1^Σ and λ_2^Σ are only slightly different, both λ_1^G and λ_2^G should be assigned relatively equal power; indeed, when $\lambda_1^\Sigma = \lambda_2^\Sigma$, the fading is uncorrelated and so, as described in Subsection 5.1.4, $\lambda_1^G = \lambda_2^G = \lambda_\mathrm{eq}^G$, which again is guaranteed by (5.31).Note that the conjecture will be validated in the following by simulations which show that the result is nearly identical to that with power allocation by exhaustive search.

Figure 5.4 illustrates the transmission rates of a cooperative system with $n_\mathrm{S} = n_\mathrm{R} = 2$ when the power allocation is carried out using the proposed algorithm. For comparison, maximum transmission rates corresponding to the benchmark are also illustrated. The figure clearly shows a good agreement between exhaustive search, i.e. the benchmark, and also the simple proposed algorithm. The difference between the proposed algorithm and the benchmark is, in fact, indistinguishable. Figure 5.4 shows the transmission rate assuming equal power allocation in the relay. Note that equal power transmission is equivalent to ignoring the knowledge of correlation at the relay. Clearly, the proposed algorithm significantly outperforms equal power transmission.

With two antennas at the relay, Figure 5.4 shows that the proposed algorithm leads to excellent results. In the next section, the proposed algorithm is extended for a system with three antennas at the relay.

5.1.6 Three antenna relay

Assuming three antennas at the relay, it is clear that according to the values of λ_1^Σ, λ_2^Σ, λ_3^Σ, P_R and P_S, capacity optimal transmission can lead to three different scenarios:

- ***Case 1:*** Transmission only via the largest eigenmode achieves capacity, i.e. transmission via the eigenvectors corresponding to λ_1^Σ, or equivalently, LER is the capacity-optimal transmission method. This scenario will occur only when (5.27) holds. In this case $\lambda_1^G > 0$ and $\lambda_2^G = \lambda_3^G = 0$.
- ***Case 2:*** Transmission only via the two largest eigenmodes achieves capacity, i.e. transmission via the eigenvectors corresponding to λ_1^Σ and λ_2^Σ, or equivalently, 2-LER is the capacity optimal transmission method. In this case $\lambda_1^G > 0$, $\lambda_2^G > 0$ and $\lambda_3^G = 0$.

- ***Case 3:*** Transmission via all three eigenmodes achieves the capacity, or equivalently, 3-LER is the capacity optimal transmission method. In this case $\lambda_1^G > 0$, $\lambda_2^G > 0$ and $\lambda_3^G > 0$.

Note that (5.27) specifies the LER-optimal region. In the following, we intend to specify necessary and sufficient conditions under which, assuming proportional power allocation, transmission only via the two largest eigenmodes in the relay (2-LER) approaches capacity. Note that proportional power allocation is motivated by the precision of the algorithm introduced in Section 5.1.5. We emphasise that the optimality of n-LER is conditioned on proportional power allocation and so it is sub-optimal, however, the results are acceptable when compared with the benchmark[4].

Conditional optimality of 2-LER

As discussed in Section 5.1.5, when LER is not optimal, transmission via the two largest eigenmodes, with proportionally assigned power, approximates the benchmark with reasonable accuracy; therefore, by setting

$$\lambda_1^G = \lambda_1^\Sigma g_2 \text{ and } \lambda_2^G = \lambda_2^\Sigma g_2 \tag{5.33}$$

we aim to derive a necessary and sufficient condition under which, the maximisation problem will be achieved by setting $\lambda_3^G = 0$ and assigning the available power in the relay, proportionally, to λ_1^G and λ_2^G.

Considering that $\lambda_1^G \geq \lambda_2^G \geq \cdots \geq \lambda_{n\mathrm{R}}^G$, one can easily conclude that if $\lambda_3^G > 0$ leads to rate loss, then all the available power in the relay has to be assigned only to λ_1^G and λ_2^G and consequently $\lambda_j^G = 0$ for $j \geq 3$. Let us assume that from the entire available power in the relay, $\epsilon > 0$ is assigned to λ_3^G and, motivated by the results of the two antenna relay scenario in Section 5.1.5, the rest of the power is proportionally distributed between λ_1^G and λ_2^G. It is easy to conclude from $\frac{\Delta C_{av}(\lambda_3^G)}{\Delta \lambda_3^G} = \frac{C_{av}(\lambda_3^G=\epsilon)-C_{av}(\lambda_3^G=0)}{\epsilon} \leq 0$ that $C_{av}(\lambda_3^G = 0) \geq C_{av}(\lambda_3^G = \epsilon)$; therefore, assigning ϵ power to λ_3^G will cause a rate loss. Now, let us assume that $\epsilon \to 0$, consequently, $\frac{\Delta C_{av}(\lambda_3^G)}{\Delta \lambda_3^G} \leq 0$ is equivalent to $\frac{\partial C_{av}(\lambda_3^G)}{\partial \lambda_3^G}|_{\lambda_3^G \to 0} \leq 0$. Therefore, $\frac{\partial C_{av}(\lambda_3^G)}{\partial \lambda_3^G}|_{\lambda_3^G \to 0} \leq 0$ specifies a region in which assigning power to λ_3^G (and consequently λ_j^G for$j \geq 3$) results in rate loss, and so one has to transmit only via λ_1^G and λ_2^G in this region.

According to the power constraint in (5.10), and assuming that $\lambda_1^G = \lambda_1^\Sigma g_2$ and $\lambda_2^G = \lambda_2^\Sigma g_2$, one can calculate the power assigned to λ_1^G and λ_2^G by calculating g_2 in (5.32) as

$$g_2 = P_2 - \alpha_2 \lambda_3^G \tag{5.34}$$

[4] Our conjecture is that the benchmark transmission rate obtained using exhaustive search is, virtually, equivalent to optimal transmission rate.

$$C(\cdot) \;=\; \log\left(1+(P_2-\alpha_2\lambda_3^G)\Big(\lambda_1^{\Sigma 2}W_1X_1+\lambda_2^{\Sigma 2}W_2X_2\Big)\right) \tag{5.37}$$

$$-\log\left(1+(P_2-\alpha_2\lambda_3^G)(\lambda_1^{\Sigma}X_1+\lambda_2^{\Sigma}X_2)+\lambda_3^G\lambda_3^{\Sigma}X_3\right). \tag{5.38}$$

$$\frac{\partial C_{av}(\lambda_3^G)}{\partial\lambda_3^G}|_{\lambda_3^G\to 0} \;=\; \mathbb{E}\Big\{\frac{\alpha_2/P_2}{1+P_2Z_2}\Big\}+\mathbb{E}\Big\{\frac{\lambda_3^{\Sigma}(1+\gamma_{\mathrm{S}}Y\lambda_3^{\Sigma})X_3}{1+P_2Z_2}\Big\} \tag{5.39}$$

$$-\mathbb{E}\Big\{\frac{\lambda_3^{\Sigma}X_3-\alpha_2(\lambda_1^{\Sigma}X_1+\lambda_2^{\Sigma}X_2)}{1+P_2(\lambda_1^{\Sigma}X_1+\lambda_2^{\Sigma}X_2)}\Big\} \tag{5.40}$$

$$=\; \frac{\alpha_2}{P_2}\mathbb{E}\{\frac{1}{1+P_2Z_2}\}+\lambda_3^{\Sigma}\mathbb{E}\{\frac{1+\gamma_{\mathrm{S}}Y\lambda_3^{\Sigma}}{1+P_2Z_2}\}-(\frac{\alpha_2}{P_2}+\lambda_3^{\Sigma})\mathcal{D}(\lambda_{1,2}^{\Sigma},P_2) \tag{5.41}$$

with

$$P_2=\frac{P_{\mathrm{R}}}{P_{\mathrm{S}}(\lambda_1^{\Sigma 2}+\lambda_2^{\Sigma 2})+N_0(\lambda_1^{\Sigma}+\lambda_2^{\Sigma})} \tag{5.35}$$

$$\alpha_2=\frac{P_{\mathrm{S}}\lambda_3^{\Sigma}+N_0}{P_{\mathrm{S}}(\lambda_1^{\Sigma 2}+\lambda_2^{\Sigma 2})+N_0(\lambda_1^{\Sigma}+\lambda_2^{\Sigma})} \tag{5.36}$$

where the index of P_2 and α_2 indicates that the 2-LER condition is being considered. Substituting (5.14) and (5.34) in (5.11) and considering that $\log(a/b)=\log(a)-\log(b)$, we obtain $C(\cdot)$ in (5.37), at the top of the next page, where $W_i=1+\gamma_{\mathrm{S}}Y\lambda_i^{\Sigma}$. The conditional optimality region of 2-LER corresponds to a region determined by $\frac{\partial C_{av}(\lambda_3^G)}{\partial\lambda_3^G}|_{\lambda_3^G\to 0}\leq 0$, that can be calculated by combining (5.37) and (5.6) which is derived in (5.39) at the top of page with

$$Z_2=\lambda_1^{\Sigma 2}W_1X_1+\lambda_2^{\Sigma 2}W_2X_2. \tag{5.42}$$

Note that the random variable X_3 at the second expectation operation on the right hand side of (5.39) is independent of Z_2 and Y, hence, X_3 can be removed in the first expectation operation because $\mathbb{E}\{X_3\}=1$. To the best of our knowledge, the first expectation and second expectations in (5.39) cannot be further simplified; however, (5.39) can be further simplified to (5.40) where

$$\mathcal{D}(\lambda_{1,2}^{\Sigma},P_2)=\frac{\Gamma(0,\zeta_1)}{P_2(\lambda_1^{\Sigma}-\lambda_2^{\Sigma})}\mathrm{e}^{\zeta_1}+\frac{\Gamma(0,\zeta_2)}{P_2(\lambda_2^{\Sigma}-\lambda_1^{\Sigma})}\mathrm{e}^{\zeta_2} \tag{5.43}$$

with $\zeta_i=(P_2\lambda_i^{\Sigma})^{-1}$. Then, the conditional optimality region of 2-LER, i.e.

$$\frac{\partial C_{av}(\lambda_3^G)}{\partial\lambda_3^G}|_{\lambda_3^G\to 0}\leq 0$$

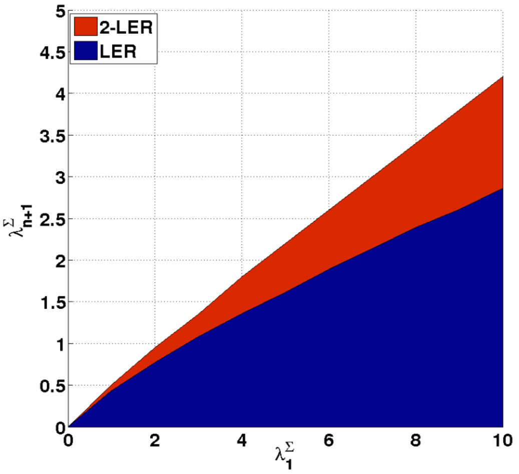


Figure 5.5: Blue area: λ_2^Σ vs. λ_1^Σ. The area shows the LER optimal region. Red area: λ_3^Σ vs. λ_1^Σ. The area illustrates the 2-LER optimal region for $\lambda_2^\Sigma = 0.5\lambda_1^\Sigma$. For both the areas $P_\mathrm{R} = P_\mathrm{S} = 10\,\mathrm{dB}$.

can be obtained by some algebraic manipulation of (5.40) according to

$$\lambda_3^\Sigma \le \frac{(\alpha_2 + P_2\lambda_3^\Sigma)\mathcal{D}(\lambda_{1,2}^\Sigma, P_2) - \alpha_2\mathbb{E}\{\frac{1}{1+P_2Z_2}\}}{P_2\mathbb{E}\{\frac{1+\lambda_3^\Sigma\gamma_\mathrm{S}Y}{1+P_2Z_2}\}} \tag{5.44}$$

. Note that when $\frac{\partial C_{av}(\lambda_3^G)}{\partial\lambda_3^G}|_{\lambda_3^G\to 0} < 0$, i.e. not including equality, the necessary condition is also sufficient, and so, the strict “inequality" of (5.44) specifies a necessary and sufficient condition, under which 2-LER is the optimal transmission method. However, when equality in (5.44) holds, i.e. meaning that $\frac{\partial C_{av}(\lambda_3^G)}{\partial\lambda_3^G}|_{\lambda_3^G\to 0} = 0$, one should make sure that the optimum point is a maximum point; this can be done by showing $\frac{\partial^2 C_{av}(\lambda_3^G)}{\partial(\lambda_3^G)^2}|_{\lambda_3^G\to 0} < 0$. In [103, App. D], it is proved that $\frac{\partial^2 C_{av}(\lambda_2^G)}{\partial(\lambda_2^G)^2}|_{\lambda_2^G\to 0} < 0$ is always valid. Following the same lines of proof, one can, similarly, prove that $\frac{\partial^2 C_{av}(\lambda_3^G)}{\partial(\lambda_3^G)^2}|_{\lambda_3^G\to 0} < 0$ and so, the inequality in (5.44) is a *necessary* and *sufficient condition* under which the 2-LER transmission is the optimal[5] transmission method.

Near optimal power allocation in a relay with three antennas

Similar to the algorithm in Section 5.1.5, introduced for power allocation in a two antenna relay, a power allocation method will be introduced for the three cases (***Case 1/2/3***) discussed at the beginning of this section.

[5] Optimal in the sense that we assume proportional power allocation to the two largest eigenvectors. In the rest of this part all n-LER transmissions are conditioned on proportional power allocation

Proposition. Allocate all available power in the relay to the largest eigenmode *if* (5.27) holds, i.e. LER, otherwise check (5.44) and allocate proportional power to the two largest eigenmodes if (5.44) holds, i.e. 2-LER; in the case when neither (5.27) nor (5.44) hold, then allocate the power proportionally to all three eigenmodes; i.e. $\lambda_1^G = \lambda_1^\Sigma g_3$, $\lambda_2^G = \lambda_2^\Sigma g_3$ and $\lambda_3^G = \lambda_3^\Sigma g_3$ where

$$g_3 = \frac{P_\mathrm{R}}{P_\mathrm{S}(\lambda_1^{\Sigma 2} + \lambda_2^{\Sigma 2} + \lambda_3^{\Sigma 2}) + N_0(\lambda_1^\Sigma + \lambda_2^\Sigma + \lambda_3^\Sigma)} \tag{5.45}$$

The algorithm is summarised in Table 5.1. ■

Figure 5.6 illustrates the transmission rates using the proposed algorithm. Comparison with the benchmark confirms that the proposed algorithm is effectively optimal.

Table 5.1: Power allocation in a relay with $n_\mathrm{R} = 3$

```
Step 1: Set $P_1 = \frac{P_\mathrm{R}}{N_0 + P_\mathrm{S}\lambda_1^\Sigma}$
Step 2: Check the inequality in (5.27)
Step 3: If Step 2 is true
            Set $\lambda_1^G = P_1$, $\lambda_2^G = \lambda_3^G = 0$ and Quit.
Step 4: Set $P_2$ from (5.35) and $\alpha_2$ from (5.36)
Step 5: Check the inequality in (5.44)
Step 6: If Step 5 is true
            Set $\lambda_1^G = \lambda_1^\Sigma g_2$, $\lambda_2^G = \lambda_2^\Sigma g_2$., $\lambda_3^G = 0$ and Quit.
        else
            Set $\lambda_1^G = \lambda_1^\Sigma g_3$, $\lambda_2^G = \lambda_2^\Sigma g_3$ and $\lambda_3^G = \lambda_3^\Sigma g_3$
```

5.1.7 Proposed power allocation in a relay with an arbitrary number of antennas

In this section, let us assume that the relay node is equipped with an arbitrary number of the antennas (say n_R). As discussed in earlier sections, in order to achieve capacity, the relay has to assign an appropriate amount of its available power per eigenvector. Following the same approach discussed in Section 5.1.6, one can assume κ cases where, depending on the the system parameters (i.e., $\mathbf{\Sigma}$, P_S and P_R), n-LER will be the capacity approaching transmission method; it means that transmission via n ($n \leq \kappa$) eigenvectors approaches capacity, and so, only n eigenvectors should be assigned power and the rest of the eigenvectors should be set to zero, i.e. $\lambda_j^G > 0$ for $j \leq n$ and $\lambda_j^G = 0$ for $j > n$.

In Section 5.1.5, we introduced the necessary and sufficient condition under which transmission via one eigenmode (largest eigenmode) achieves capacity; later on, in Section 5.1.6, a necessary and sufficient condition was derived, under which, transmission via the two largest eigenmodes approaches maximum transmission rate. One can extend the same

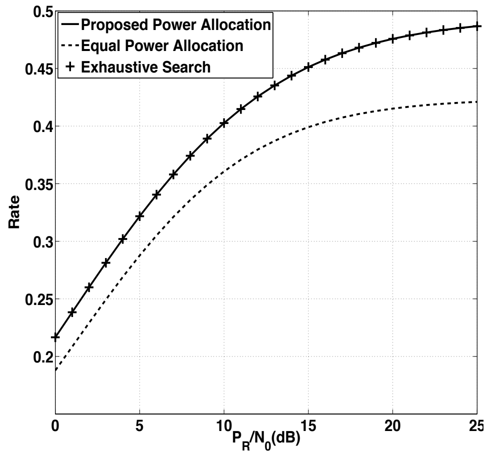


Figure 5.6: Rate vs. transmit power at the relay. The rate values are obtained using the proposed power allocation algorithm, equal power allocation and the benchmark. $n_{\mathrm{S}} = 2$, $n_{\mathrm{R}} = 3$, $N_0 = 1$, $P_{\mathrm{S}} = 0\,\mathrm{dB}$, and inter-antenna correlation $\rho_{12} = 0.7$, $\rho_{23} = 0.5$ and $\rho_{13} = 0.2$. The exhaustive search is performed over discrete values of transmission rates that were obtained using various values of λ_1^G and λ_2^G (step size 0.1 dB) that fulfil (5.10).

concept and derive a necessary and sufficient condition under which transmission via n ($n \leq \kappa$) eigenvectors will maximise the rate with the assumption of proportional power allocation.

Analogous to Section 5.1.6, let us assume that from the available power in the relay, $\epsilon > 0$ is assigned to λ_{n+1}^G and the rest of the power is proportionally distributed between $\lambda_1^G \cdots \lambda_n^G$. It is easy to conclude that $\frac{\partial C_{av}(\lambda_{n+1}^G)}{\partial \lambda_{n+1}^G}|_{\lambda_{n+1}^G \to 0} \leq 0$ is equivalent to the fact that assigning power to λ_{n+1}^G will result in rate loss, and consequently, since $\lambda_1^G \geq \lambda_2^G \geq \cdots \lambda_{n_{\mathrm{R}}}^G$, the egienvectors corresponding to λ_j^Σ for $j \geq n+1$ should not be assigned power.

Theorem 5.4. *The necessary and sufficient condition under which transmission from n largest eigenmodes (n-LER transmission) approaches capacity is:*

$$\lambda_{n+1}^\Sigma \leq \frac{(\alpha_n + P_n \lambda_{n+1}^\Sigma)\mathcal{D}(\lambda_{1,\cdots,n}^\Sigma, P_n) - \alpha_n \mathbb{E}\{\frac{1}{1+P_n Z_n}\}}{P_n \mathbb{E}\{\frac{1+\lambda_{n+1}^\Sigma \gamma_S Y}{1+P_n Z_n}\}} \tag{5.46}$$

where

$$P_n = \frac{P_R}{N_0 \sum_{m=1}^n \lambda_m^\Sigma + P_S \sum_{m=1}^n \lambda_m^{\Sigma 2}} \tag{5.47}$$

$$\alpha_n = \frac{N_0 + P_S \lambda_{n+1}^\Sigma}{N_0 \sum_{m=1}^n \lambda_m^\Sigma + P_S \sum_{m=1}^n \lambda_m^{\Sigma 2}} \tag{5.48}$$

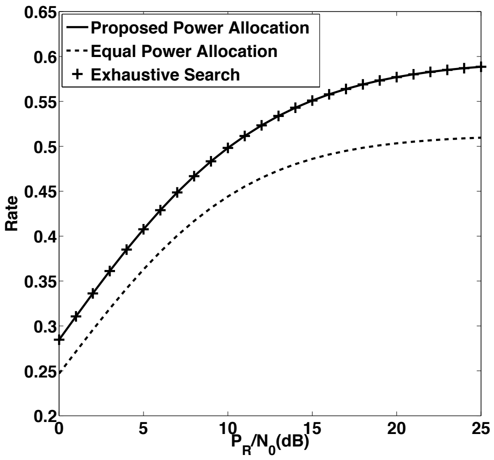


Figure 5.7: Rate vs. transmit power at the relay. The rate values are obtained using the proposed power allocation algorithm, equal power allocation and the benchmark. $n_\mathrm{S} = 2$, $n_\mathrm{R} = 4$, $N_0 = 1$, $P_\mathrm{S} = 0\,\mathrm{dB}$, and inter-antenna correlation $\rho_{12} = 0.7$, $\rho_{13} = 0.5$, $\rho_{14} = 0.3$, $\rho_{23} = 0.7$, $\rho_{24} = 0.5$ and $\rho_{34} = 0.7$. The exhaustive search is performed over discrete values of transmission rates that were obtained using various values of λ_i^G (step size 0.1 dB) that fulfil (5.10).

and

$$\mathcal{D}(\lambda^{\Sigma}_{1,\cdots,n}, P_n) = \sum_{m=1}^{n} \frac{(\lambda^{\Sigma}_m)^{n-2}}{\prod\limits_{\substack{k=1\\k\neq m}}^{n} (\lambda^{\Sigma}_m - \lambda^{\Sigma}_k)} \Gamma(0, \zeta_m) \mathrm{e}^{\zeta_m} \tag{5.49}$$

Proof. The proof is similar to that of Section 5.1.6. □

Note that (5.46) is a general form of (5.27) and (5.44). In fact, (5.46) will determine the eigenvectors that should be assigned power proportional to their correlation power. The power allocation algorithm is summarised in Table 5.2. The algorithm is indeed analogous to the water-filling algorithm: the purpose is to find the eigenmodes that should be assigned power for transmission and to discard the “weak" eigenmodes that result in rate-loss if assigned power. Note that the expressions derived in (5.27), (5.44) and (5.46) involve expectation operations that do not seem to lend themselves to calculation in closed-form; therefore, in practical implementation of the system one should implement them using numerical methods. Please see [103] wherein a numerical integration expression is derived.

Figure 5.7 illustrates the transmission rates for the proposed algorithm for a relay with four antennas and correlation coefficients as described in the caption. Clearly the proposed algorithm agrees with the rates obtained using exhaustive search. Moreover, its superiority over equal power transmission is evident. The values chosen for inter-antenna correlation in the numerical simulations in Figures 5.6 and 5.7 are selected to be relatively high, and to reflect what might be expected in a relatively closely-spaced linear array. In these cases the proposed algorithm demonstrates performance extremely close to the optimum. Note that when $\rho_{ij} \rightarrow 0$ (smaller inter-antenna correlation), the spatial correlation diminishes and so the equal power transmission will be the optimum method, which is indeed guaranteed by the proportional power allocation proposed in (5.31); this is demonstrated by numerical simulation in Figure 5.4. Hence, since our algorithm is provably optimum at low correlation and is shown by simulation to be very close to optimum for a typical case of high correlation, it is at least a reasonable hypothesis that it is near optimum for all cases of practical interest.

Table 5.2: Power allocation in a relay with arbitrary n_{R}

```
Initiate $n = 1$
while $n \leq \kappa$
Step 1: Set $P_n$ from (5.47) and $\alpha_n$ from (5.48)
Step 2: Check the inequality in (5.46)
Step 3: If Step 2 is true
    Set $\lambda_j^G = \lambda_j^\Sigma g_n$ for $j \leq n$
    Set $\lambda_j^G = 0$ for $j > n + 1$ and Quit while.
else
    Set $n \leftarrow n + 1$ and go to Step 1
end (end of while when $n > \kappa$)
```

5.1.8 Conclusion

This part studies the statistical characteristics of the received SNR at the destination in a MIMO relay network when the relay node experiences fading correlation. It is assumed that the relay node has access only to the statistical CSI. In order to approach the ergodic capacity of the system, based on the available statistical channel knowledge at the relay, a new relay precoder design methodology is introduced. The proposed method is analogous to the water-filling algorithm; it searches for the largest eigenmodes that should be assigned power and discards the remaining eigenmodes. The simulations demonstrate good agreement between the proposed method and the benchmark which is obtained using exhaustive search.

5.1.9 Proof of the SNR distribution: Full correlation

By combining (5.14) and (5.17), one can write $\gamma_{\mathrm{D}} = \gamma_{\mathrm{S}} n_{\mathrm{R}} \frac{YV}{1+V}$, and so, assuming $w = \frac{x}{\gamma_{\mathrm{S}} n\mathrm{R}}$ we have

$$
\begin{aligned}
F_{\gamma_{\mathrm{D}}-FC}(x) &= \mathbb{P}(\frac{YV}{1+V} < w) = \mathbb{P}(V < \frac{w}{Y-w}) \\
1 - \mathrm{e}^{\frac{-w}{\lambda_1^G n_{\mathrm{R}}(Y-w)}} &= \int_0^\infty (1 - \mathrm{e}^{\frac{-w}{\lambda_1^G n_{\mathrm{R}}(y-w)}}) f_Y(y) \mathrm{d}y \\
& 1 - \int_0^\infty \mathrm{e}^{\frac{-w}{\lambda_1^G n_{\mathrm{R}}(y-w)}} f_Y(y) \mathrm{d}y
\end{aligned}
\tag{5.50}
$$

Note that V is exponentially distributed, and so, $\mathbb{P}(V < \frac{w}{Y-w})$ is non-zero only for $Y > 0$; consequently, by substituting $f_Y(y)$ in (5.50) and assuming the proper domain for the integral, we have

$$
F_{\gamma_{\mathrm{D}}-FC}(x) = 1 - \frac{n_{\mathrm{S}}^{n_{\mathrm{S}}}}{(n_{\mathrm{S}}-1)!} \int_w^\infty y^{n_{\mathrm{S}}-1} \mathrm{e}^{-\frac{w}{\lambda_1^G n_{\mathrm{R}}(y-w)} - n_{\mathrm{S}} y} \mathrm{d}y \tag{5.51}
$$

by changing variable according to $t = y - w$, assuming

$$
(t+w)^{n_{\mathrm{S}}-1} = \sum_{m=0}^{n_{\mathrm{S}}-1} \frac{(n_{\mathrm{S}}-1)!}{m!(n_{\mathrm{S}}-m-1)!} t^m w^{n_{\mathrm{S}}-m-1}
$$

and applying [113, 3.471.9], $F_{\gamma_{\mathrm{D}}-FC}(x)$ will be derived as

$$
F_{\gamma_{\mathrm{D}}-FC}(w) = 1 - 2(n_{\mathrm{S}} w)^{n_{\mathrm{S}}} \mathrm{e}^{-n_{\mathrm{S}} w} \sum_{m=0}^{n_{\mathrm{S}}-1} \frac{(\lambda_1^G n_{\mathrm{S}} n_{\mathrm{R}} w)^{-(m+1)/2}}{m!(n_{\mathrm{S}}-m-1)!} \mathrm{K}_{m+1}(2\sqrt{\frac{n_{\mathrm{S}} w}{\lambda_1^G n_{\mathrm{R}}}})
$$

and so, (5.18) is proved.

5.1.10 Statistics of the random variable X

As a complete proof of (5.20) is lengthy, we only prove the case of $\kappa = 2$. Following the same approach, the extension of the proof to larger values of κ, i.e. $\kappa = 3, 4, \cdots, n$, is straightforward. Then by the rule of mathematical induction, it is easy to obtain the general expression in (5.20).

$F_X(x)$ **for the case of** $\kappa = 2$. The numerator and the denominator random variable X in (5.16) include summation of random variables X_j which follow exponential distribution with unit mean. For simplicity, we assume $\kappa = 2$, i.e. $X = \frac{\lambda_1^G \lambda_1^{\Sigma 2} X_1 + \lambda_2^G \lambda_2^{\Sigma 2} X_2}{1 + \lambda_1^G \lambda_1^{\Sigma} X_1 \lambda_2^G \lambda_2^{\Sigma} X_2}$, and derive $F_X(x)$ of (5.20). However, in order to simplify notation, in this section, let us substitute $\lambda_j^G X_j \rightarrow X_j$ consequently X_j is distributed exponentially with mean λ_j^G. One can write

$$F_X(x) = \mathbb{P}(X < x) = \mathbb{P}\big(\frac{\lambda_1^{\Sigma 2} X_1 + \lambda_2^{\Sigma 2} X_2}{1 + \lambda_1^{\Sigma} X_1 + \lambda_2^{\Sigma} X_2} < x\big). \tag{5.52}$$

Applying basic algebraic manipulation, (5.52) can be simplified according to

$$\mathbb{P}(X < x) \ \triangleq \ \mathbb{P}\big(X_1 < \frac{x + \lambda_2^{\Sigma}(x - \lambda_2^{\Sigma}) X_2}{\lambda_1^{\Sigma}(\lambda_1^{\Sigma} - x)}\big) \tag{5.53}$$

Let us split the problem of deriving $\mathbb{P}(X < x)$ in (5.53) over four different intervals: ***A)*** $x < 0$, ***B)*** $0 < x < \lambda_2^{\Sigma}$ ***C)*** $\lambda_2^{\Sigma} < x < \lambda_1^{\Sigma}$ and ***D)*** $\lambda_1^{\Sigma} < x$ and derive $\mathbb{P}(X < x)$ for each interval individually.

A) $x < 0$**.** Note that the random variables X, X_1 and X_2 are non-negative random variables. Therefore, for $x < 0$, we have $\mathbb{P}(X < x) = 0$.

B) $0 < x < \lambda_2^{\Sigma}$**.** Assuming $0 < x < \lambda_2^{\Sigma}$; clearly, $(\lambda_1^{\Sigma} - x)$ at the denominator of (5.53) is positive. Therefore, since $X_1 > 0$, the expression $x + \lambda_2^{\Sigma}(x - \lambda_2^{\Sigma}) X_2$ at the numerator of (5.53) has to be positive as well and, hence,

$$X_2 < \frac{-x}{\lambda_2^{\Sigma}(x - \lambda_2^{\Sigma})} \tag{5.54}$$

must hold. Therefore, (5.53) can be simplified to

$$\mathbb{P}(X < x) = \int_0^{\frac{-x}{\lambda_2^{\Sigma}(x - \lambda_2^{\Sigma})}} \mathbb{P}\big(X_1 < \frac{x + \lambda_2^{\Sigma}(x - \lambda_2^{\Sigma}) x_2}{\lambda_1^{\Sigma}(\lambda_1^{\Sigma} - x)}\big) \underbrace{\frac{\mathrm{e}^{-x_2/\lambda_2^G}}{\lambda_2^G}}_{f_{X_2}(x_2)} \mathrm{d}x_2.$$

Considering that $\mathbb{P}(X_1 < t) = 1 - \mathrm{e}^{-t/\lambda_2^G}$, it is straightforward to prove, from (5.55), that for $0 < x < \lambda_2^{\Sigma}$,

$$\mathbb{P}(X < x) = 1 - \frac{c_2}{c_2 - c_1} \mathrm{e}^{c_1 x} - \frac{c_1}{c_1 - c_2} \mathrm{e}^{c_2 x}. \tag{5.55}$$

where c_j is defined in (5.21).

C) $\lambda_2^\Sigma < x < \lambda_1^\Sigma$**.** It is easy to deduce from (5.53) that for $\lambda_2^\Sigma < x < \lambda_1^\Sigma$, the numerator and the denominator of (5.53) is positive for every value of the random variable X_2 and, hence,

$$\mathbb{P}(X < x) = \int_0^\infty \mathbb{P}\big(X_1 < \frac{x + \lambda_2^\Sigma(x - \lambda_2^\Sigma)x_2}{\lambda_1^\Sigma(\lambda_1^\Sigma - x)}\big)\frac{\mathrm{e}^{-x_2/\lambda_2^G}}{\lambda_2^G}\mathrm{d}x_2 \tag{5.56}$$

which leads to

$$\mathbb{P}(X < x) = 1 - \frac{c_2}{c_2 - c_1}\mathrm{e}^{c_1 x} \tag{5.57}$$

D) $x > \lambda_1^\Sigma$**.** For the given interval, clearly, the denominator of (5.53) is negative while the numerator is positive; hence, one can easily conclude that assuming $x > \lambda_1^\Sigma$ is equivalent to assuming $X_1 < 0$. On the other hand, since X_1 follows the exponential distribution, it is necessarily positive, and so, $x > \lambda_1^\Sigma$ is an invalid assumption. Consequently, $\mathbb{P}(X < x) = 1$ for $x > \lambda_1^\Sigma$.

Part III

Signal processing and decoding strategies

6 Hierarchical interference processing

6.1 Successive decoding with hierarchical interference cancellation in wireless (physical layer) network coding systems

The intermediate network nodes (relays) in WPLNC systems are allowed to process only some specific (“hierarchical") functions of user data (instead of separate decoding of individual user signals), which allows to overcome the rate limits induced by the conventional Multiple Access Channels capacity regions. Unfortunately, since the mapping between a specific decoded hierarchical function and the corresponding signal space representation is generally *one-to-many*, traditional *successive decoding techniques* with perfect interference cancellation (subtraction) of decoded signals cannot be directly employed in WPLNC systems. In this section we show that even in this case the knowledge of hierarchical data can be efficiently exploited in the decoding process to significantly *reduce the impact of interfering signal* on the subsequent decoding operations in WPLNC systems.

6.1.1 Introduction

Background and related work

The invention of network coding [37, 44] has provided a valuable tool to boost the performance of communication networks beyond the conventional routing-based solutions. It is not surprising that wireless systems researchers soon turned their attention to this powerful wireline technique too.

Even though that the conventional network coding techniques can be (relatively straightforwardly) implemented in wireless communication systems, it was realised later that wireless channels possess some favourable characteristics, e.g. inherent broadcast and superposition nature, which can be efficiently exploited together with a modified network coding processing [24, 47]. The techniques based on such extension of conventional network coding to the wireless domain are usually referred to as WPLNC [1, 11, 21, 22].

In WPLNC systems the intermediate network nodes, e.g. relays, do not have to decode separate source data from their observations, but instead of this, they directly decode *hierarchical functions* of source data, e.g. $\mathbf{d}_{AB} = f(\mathbf{d}_A, \mathbf{d}_B)$, allowing to achieve the rates above the conventional Multiple-Access (MAC) capacity region, see [1, 22].

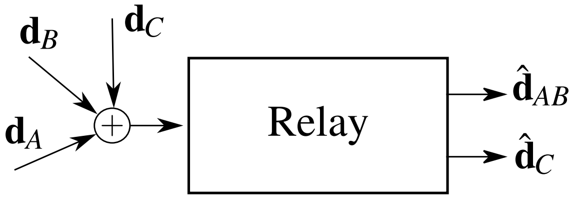


Figure 6.1: WPLNC relay receiving transmissions from three independent sources.

Analyses of WPLNC processing in wireless systems have, however, revealed several non-trivial research problems which do not appear in conventional wireless systems, including the sensitivity to channel parameterisation [7–9, 25], challenging multi-source transmission synchronisation [27–29] or partial decoding of information on broadcast channels [4, 115, 116].

Goals of this report and contribution

Another interesting research problem occurs when the relay observation contains (apart of $\mathbf{d}_A, \mathbf{d}_B$) another source data, e.g. $\mathbf{d}_C$), which cannot be separately decoded from the relay observation (see Figure 6.1. In a conventional MAC channel, *Successive Decoding (SD) with Interference Cancellation (IC)* is used as an optimal decoding strategy, allowing to achieve the rates at the upper-bound of MAC capacity region [32]. The possibility to perform perfect IC is there guaranteed by a *one-to-one mapping* between the decoded source data and their signal space representation. Unfortunately, only specific functions of source data are decoded by relays in WPLNC systems, and hence it is impossible to perform IC (subtraction) of the decoded signal from the relay observations, as the mapping between the decoded function of source data and their signal space representation is generally *one-to-many* (see Figure 6.2).

An alternative approach to multi-user WPLNC decoding is introduced in [117], where an approximate sum-capacity is evaluated for a lattice-encoded, K-user interference channel. The decoding process presented in [117] is based on a decoding[1] of multiple (hierarchical) functions of source signals at each receiving node, which can be then solved to obtain the desired data. However, for the present, the system model analysed in [117] is limited to a *real-valued* interference channel case with *equal strength interferers*, and hence it is not able to capture the impact of complex channel parameterisation or different interferer strengths.

In this section we discuss the *successive-decoding* problem in WPLNC systems. We introduce a 3-user single-relay network topology, which is capable to illustrate a general background of the problem. We exemplify the relay decoding process in a binary-modulated

[1] The decoding operation is based on a modulo-lattice operation in the Compute and Forward [22] framework.

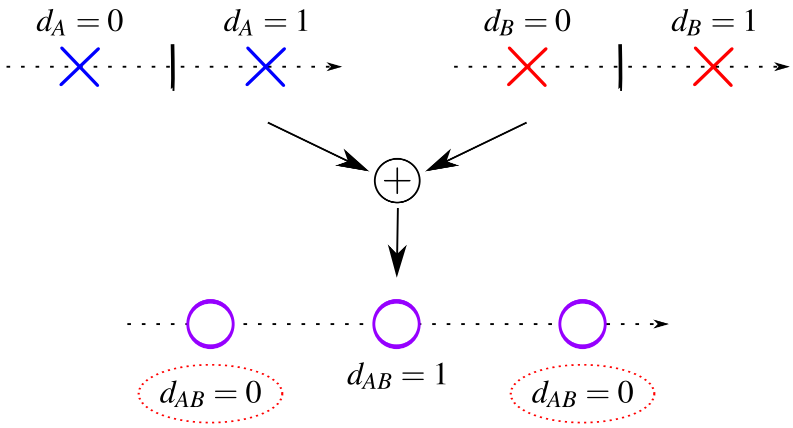


Figure 6.2: Multiple constellation symbols correspond to the decoded hierarchical data symbol $d_{AB} = 0$ (binary modulated sources A, B, $d_{AB} = d_A \oplus d_B$).

system, where we show that a knowledge of $\mathbf{d}_{AB}$ (more precisely $\hat{\mathbf{c}}_{AB}(\mathbf{d}_{AB})$) is sufficient to enable the subsequent decoding of $\mathbf{d}_C$ to be (under some conditions) almost *interference free*, even though that perfect IC cannot be performed. We show that even in a binary-modulated system, varying parameterisations of particular wireless channels can significantly affect the achievable performance. We do not strive to evaluate the capacity of the system as a function of channel parameterisation, but rather we would like to demonstrate that *successive decoding* with *hierarchical interference cancellation* represents a viable decoding strategy for multi-user WPLNC systems.

6.1.2 System model and definitions

The simplest network topology where the principle of successive decoding in WPLNC systems can be demonstrated is a *relay MAC channel with three independent sources* (Figure 6.3). This topology can be viewed as some inner building block of a large wireless network.

In the analysed system, three sources (A, B, C) simultaneously transmit their data to a common shared relay (R). A signal space representation (with an *orthonormal* basis) of transmitted channel symbols is $s_i(c_i) \in \mathcal{A}_s^i \subset \mathbb{C}^N$, $i \in \{A, B, C\}$, where $\mathcal{A}_s^i(\centerdot)$ is a channel symbol memoryless mapper, $\mathbf{c}_i(\mathbf{d}_i)$ is a source i codeword and $\mathbf{d}_i$ is a data-word sent from the source i.

The n-th constellation space symbol (we omit the time variable n for a better clarity) received at the relay is

$$y_R = h_A s_A + h_B s_B + h_C s_C + w \tag{6.1}$$

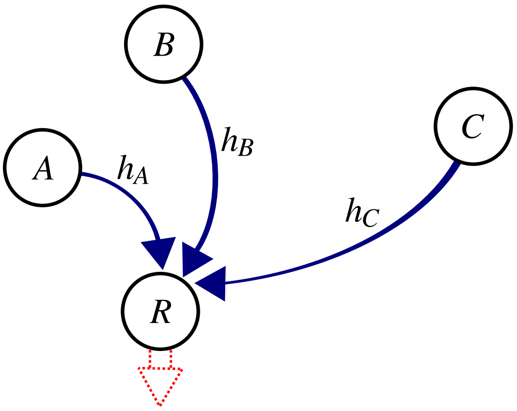


Figure 6.3: Relay MAC channel with three independent sources A, B, C.

where w is the circularly symmetric complex Gaussian noise (variance σ_w^2 per complex dimension) and h_A, h_B, h_C are scalar complex channel coefficients (constant during the observation and known at the relay).

Without loss of generality we assume that the relay wants to recover only some function of data from sources A, B ($\mathbf{d}_{AB} = f\left(\mathbf{d}_{A,}\mathbf{d}_B\right)$), while data from source C ($\mathbf{d}_C$) has to be decoded separately. These two decoded data streams ($\mathbf{d}_{AB}$, $\mathbf{d}_C$) are then potentially re-encoded by the relay and transmitted to the network. In this report we are interested only in the relay decoding process, and hence we will not discuss the particular form of the relay output signal.

6.1.3 Overview of successive decoding strategies in WPLNC systems

As we have already mentioned, successive decoding with interference cancellation (SD-IC) is well understood to be a capacity achieving decoding strategy in conventional multi-user MAC channels. The main principle of SD-IC resides in the fact that the decoded signal can be perfectly subtracted from the receiver observation, thus removing the corresponding interfering signal from the subsequent receiver processing. Unfortunately, this is not always possible in WPLNC systems, due to the *one-to-many mapping* between the decoded function of source data and its signal space representation (see Figure 6.2).

In this section we overview three *successive decoding strategies* for the analysed MAC channel (Figure 6.3). We show that only two of these decoding strategies can be implemented using conventional SD-IC processing, while the last one does not allow a perfect IC of the decoded signal, and hence it introduces a novel research problem.

Conventional single-user decoding

Single-user decoding of separate user data streams $\mathbf{d}_A$, $\mathbf{d}_B$, $\mathbf{d}_C$ is evidently a feasible strategy, even though the fundamental goal of the relay node is to forward $\mathbf{d}_{AB}$, $\mathbf{d}_C$ to the network[2]. A conventional *3-user MAC capacity region* [32] then defines the upper-bound of source transmission rates, and an arbitrary rate 3-tuple within this capacity region can be achieved using a standard SD-IC processing. However, constraints given by the MAC capacity region does not allow to fully exploit the available performance gains of WPLNC processing.

WPLNC-based decoding

WPLNC processing allows to overcome some constraints induced by conventional MAC capacity regions, as only some functions of data are decoded at intermediate network nodes (instead of separate decoding of particular user data). As we show in the following discussion, the order in which the signals are decoded determines the possibility to perform a *perfect IC of decoded signals* and consequently also the achievable performance.

Perfect IC of s_C. We assume that source transmission rates allow that only $\mathbf{d}_C$ can be decoded from the relay observation (6.1), while $\mathbf{d}_{AB}$ cannot be directly decoded. Since mapping between the code symbol $c_C\left(d_C\right)$ and its signal space representation s_C is one-to-one, a perfect IC can be performed, and hence the influence of s_C can be completely removed from the relay observation:

$$\begin{aligned} y_R' &= y_R - h_C s_C\left(c_C\right) \\ &= h_A s_A\left(c_A\right) + h_B s_B\left(c_B\right) + w. \end{aligned} \tag{6.2}$$

This allows an *interference-free decoding* of $\mathbf{c}_{AB}\left(\mathbf{d}_{AB}\right)$ from (6.2). Thus, after IC the system becomes equivalent to a conventional *2-Way Relay Channel* (2-WRC) with WPLNC decoding of $\mathbf{d}_{AB}$ [1]. Since the perfect IC is again possible, we will not further analyse this particular decoding strategy here.

Hierarchical IC (Imperfect IC of $h_A s_A + h_B s_B$). We assume that source transmission rates allow that only $\mathbf{d}_{AB}$ can be decoded from the relay observation (6.1), while $\mathbf{d}_C$ cannot be directly decoded. Since mapping between the hierarchical code symbol c_{AB} and its signal space representation (signal $h_A s_A\left(c_A\left(d_A\right)\right) + h_B s_B\left(c_B\left(d_B\right)\right)$) is one-to-many, perfect IC cannot be performed. However, even in this case the knowledge of

[2] Note that $\mathbf{d}_{AB} = f\left(\mathbf{d}_A, \mathbf{d}_B\right)$ can be always obtained when both $\mathbf{d}_A$, $\mathbf{d}_B$ are available at the relay.

$\mathbf{c}_{AB}\left(\mathbf{d}_{AB}\right)$ can be efficiently exploited by the Maximum Likelihood decoder of $\mathbf{c}_C\left(\mathbf{d}_C\right)$:

$$\begin{aligned}
\hat{\mathbf{c}}_C &= \arg\max_{\check{\mathbf{c}}_C} p\left(\mathbf{y}_R \mid \check{\mathbf{c}}_C, \hat{\mathbf{c}}_{AB}\right) \\
&\overset{(*)}{\approx} \arg\max_{\check{\mathbf{c}}_C} \prod_n p\left(y_R \mid \check{c}_C, \hat{c}_{AB}\right) \\
&= \arg\max_{\check{\mathbf{c}}_C} \prod_n \sum_{c_A, c_B : f'(c_A, c_B) = \hat{c}_{AB}} p\left(y_R \mid \check{c}_C, c_A, c_B\right) \\
&= \arg\max_{\check{\mathbf{c}}_C} \prod_n \sum_{c_A, c_B : f'(c_A, c_B) = \hat{c}_{AB}} \exp\left(\frac{-\left\|y''_R - h_C s_C\left(\check{c}_C\right)\right\|^2}{\sigma_W^2}\right)
\end{aligned} \tag{6.3}$$

where $y''_R = y_R - h_A s_A\left(c_A\right) - h_B s_B\left(c_B\right)$ and the approximation in $(*)$ is given by the assumption of *per-symbol evaluated metric*[3]. Without loss of generality we have assumed that the hierarchical function can be evaluated element-wisely on code symbols, i.e. $c_{AB} = f'\left(c_A, c_B\right)$.

Contrary to the previous case, the perfect IC cannot be performed since only a function of source data $\mathbf{d}_{AB} = f\left(\mathbf{d}_{A,}\mathbf{d}_B\right)$ (equivalently a function of source codewords $\mathbf{c}_{AB} = f'\left(\mathbf{c}_{A,}\mathbf{c}_B\right)$) is decoded, and hence the exact value of the corresponding signal space representation $h_A s_A\left(c_A\right) - h_B s_B\left(c_B\right)$ remains unknown to the relay. Nevertheless, a knowledge of a particular subset of codewords ($\mathbf{c}_A$, $\mathbf{c}_B$) conforming with $\mathbf{c}_{AB} = f'\left(\mathbf{c}_A, \mathbf{c}_B\right)$ becomes available at the relay when $\hat{\mathbf{c}}_{AB}\left(\mathbf{d}_{AB}\right)$ is decoded, and hence at least a specific subset of permissible values of $h_A s_A\left(c_A\right) - h_B s_B\left(c_B\right)$ for a particular c_{AB} can be identified. Presumably, a knowledge of this subset can help to reduce the impact of the interfering signal $h_A s_A\left(c_A\right) - h_B s_B\left(c_B\right)$ on the relay processing. Since the knowledge of *hierarchical codewords* $\mathbf{c}_{AB} = f'\left(\mathbf{c}_{A,}\mathbf{c}_B\right)$ is exploited in the decoding process, we will refer to this decoding strategy as the *"Successive Decoding with Hierarchical Interference Cancellation"* (SD-HIC).

6.1.4 Binary system example

In the rest of this report we analyse the performance of SD-HIC strategy (6.3) in a BPSK-modulated WPLNC system, where $s_i\left(c_i\right) \in \mathcal{A}_s^i = \{\pm 1\}, i \in \{A, B, C\}$. The hierarchical function f' is defined on codewords as

$$f'\left(\mathbf{c}_A, \mathbf{c}_B\right) = \mathbf{c}_A \oplus \mathbf{c}_B, \tag{6.4}$$

where $\oplus$ is a bit-wise xor operation.

[3] The per-symbol evaluation of metric could be sub-optimal since $p\left(y_R^{(n)} \mid \check{c}_C^{(n)}, \hat{c}_{AB}^{(n)}\right)$ and $p\left(y_R^{(n')} \mid \check{c}_C^{(n')}, \hat{c}_{AB}^{(n')}\right)$ are not generally independent for $n \neq n'$.

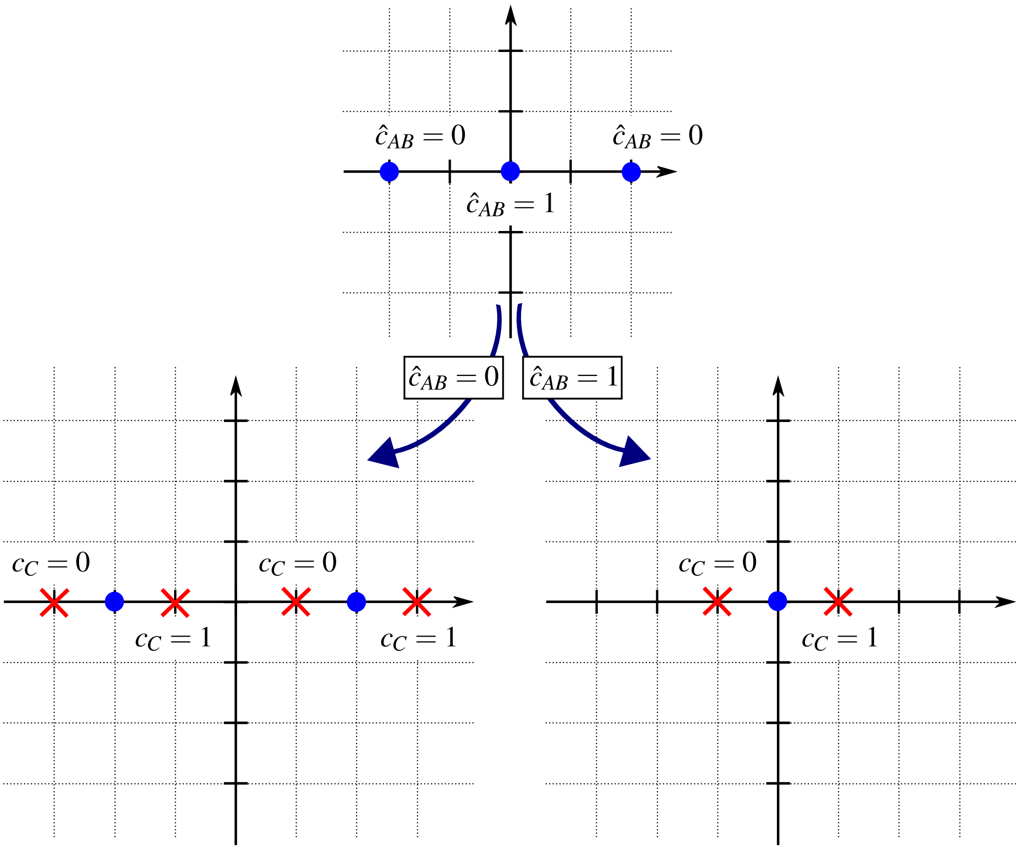


Figure 6.4: Relay observation in SD-HIC process: Pparameterisarameterisation case I ($\angle h_A = \angle h_B = \angle h_C$ and $|h_i| = 1$).

First of all, we focus on the decoding of source C data in a genie-aided relay decoder which has a perfect estimate of hierarchical codewords $\hat{\mathbf{c}}_{AB}$ $(\mathbf{d}_{AB})$ available[4], and hence only a subsequent decoding of $\mathbf{c}_C$ $(\mathbf{d}_C)$ takes place. We depict the relay observation in SD-HIC process to get a better insight into the problem and we analyse three cases of *channel parameterisations* to emphasise its impact on the processing. Then, we provide a numerical evaluation of *capacity* (mutual information) performance of the BPSK system to demonstrate that HIC is capable to reduce the impact of source interference, even though that perfect elimination of interfering signal is impossible.

Impact of channel parameterisation

The knowledge of a particular value of hierarchical code symbol c_{AB} virtually splits the relay observation in SD-HIC process into several sub-cases, e.g. for $\hat{c}_{AB} = 0$ and $\hat{c}_{AB} = 1$ in the BPSK-modulated system. As a result, there are potentially multiple signal space

[4] This corresponds to the situation where source rates allow an error-free decoding $\hat{\mathbf{c}}_{AB}$ $(\mathbf{d}_{AB})$ from the relay observation in the presence of interfering signal from source C.

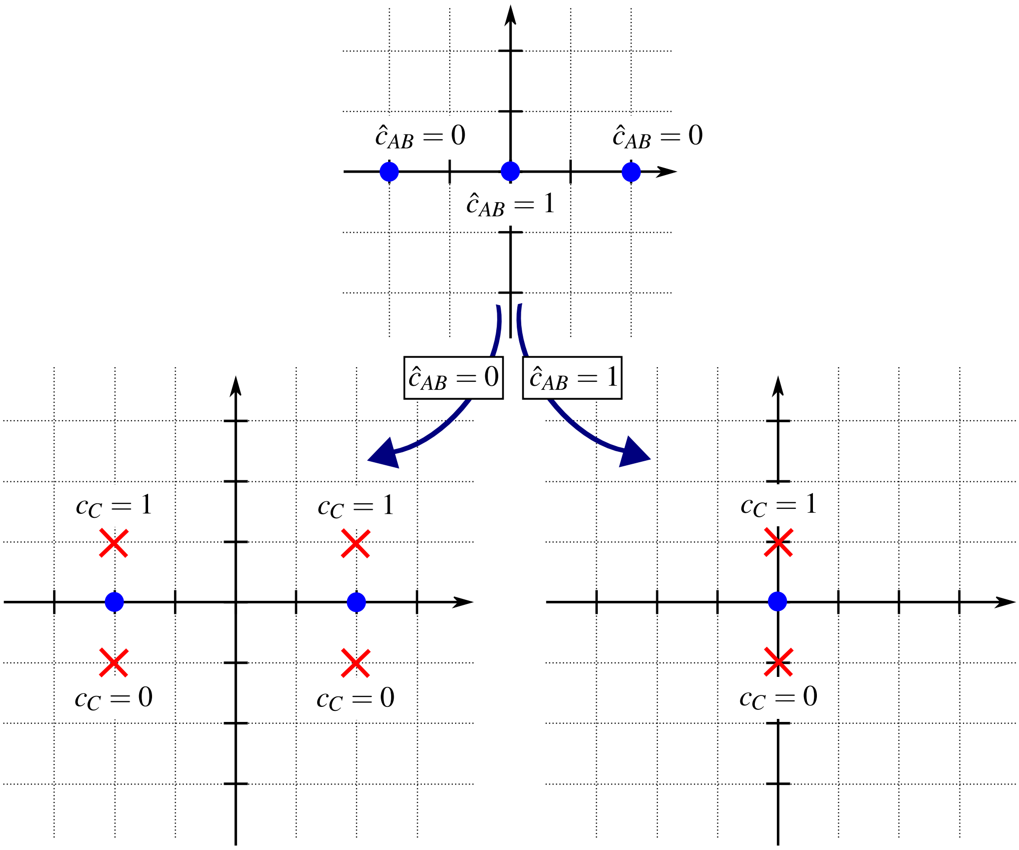


Figure 6.5: Relay observation in SD-HIC process: parameterisation case II ($\angle h_A = \angle h_B = 0$; $\angle h_C = {}^{\pi}/_{2}$ and $|h_i| = 1$).

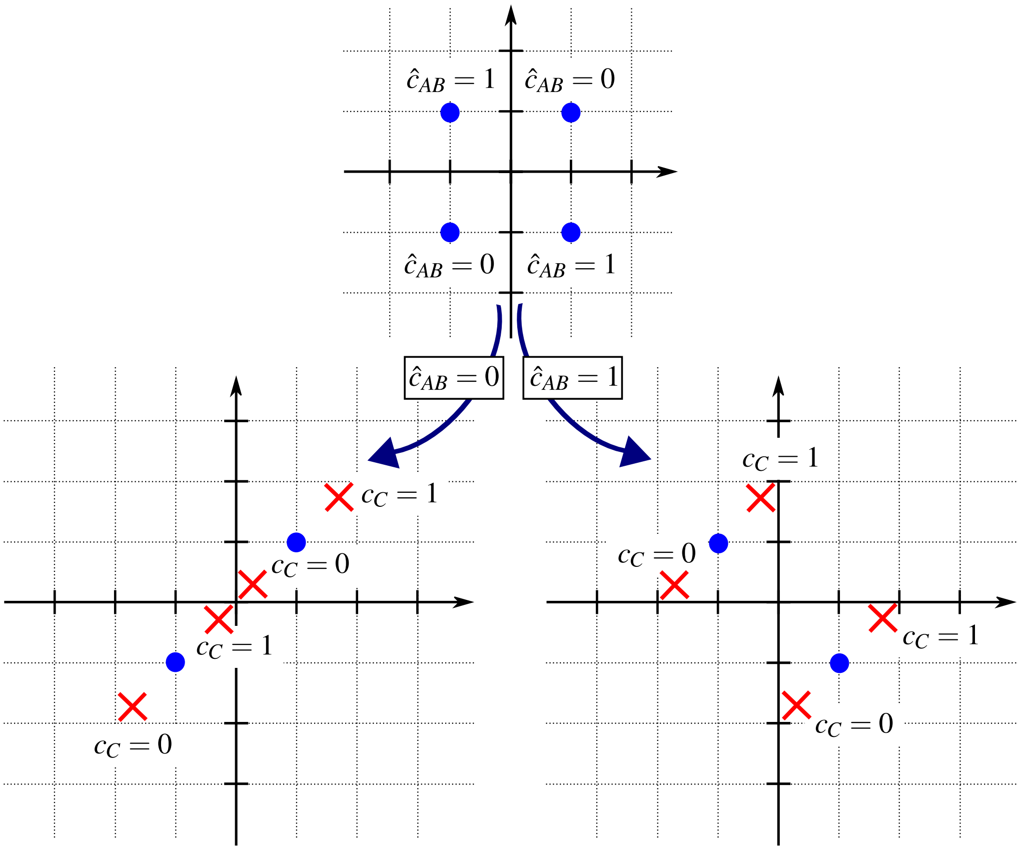


Figure 6.6: Relay observation in SD-HIC process: Parameterisation case III ($\angle h_A = 0$; $\angle h_B = \pi/2$; $\angle h_C = \pi/4$ and $|h_i| = 1$).

points conforming to a particular code symbol c_C. As we show in the following examples, the particular number of these signal space points depends mainly on the actual channel parameterisation and particular value of c_{AB}.

Parameterisation case I: $\angle h_A = \angle h_B = \angle h_C$**.** At first we analyse the case when all source-relay channels are synchronous in phase, i.e. $\angle h_A = \angle h_B = \angle h_C$. As it is visualised in Figure 6.4, the knowledge of c_{AB} virtually splits the relay observation in SD-HIC process into two sub-cases for each particular signal space symbol observation.

Parameterisation case II: $\angle h_A = \angle h_B = 0$; $\angle h_C = {}^{\pi}/_{2}$**.** As a second example we analyse the case where signals from sources A, B are synchronous in phase, while the signal from source C is rotated by $\angle h_C = {}^{\pi}/_{2}$. As it is visualised in Figure 6.5, in this case the signal from source C appears to be virtually orthogonal to signals from sources A, B. Presumably, in this case the impact of imperfect interference elimination on the performance of the system will be almost negligible. However, it is important to note that this virtual orthogonality of signals can presumably occur only if all the sources are BPSK-modulated.

Parameterisation case III: $\angle h_A = 0$; $\angle h_B = {}^{\pi}/_{2}$; $\angle h_C = {}^{\pi}/_{4}$**.** As a last example we analyse the case where all source signals are asynchronous in phase. As it is visualised in Figure 6.6, under some specific channel phase rotation, constellation symbols conforming to a different code symbols ($c_C = 0$, $c_C = 1$) can fall close to each other in the constellation space, which will presumably result in a decreased performance of SD-HIC processing. Note that similar behaviour (Euclidean distance shortening due to an undesirable channel parameterisation) has been already observed in hierarchical signal decoders, see [8].

Reference decoders

In this report we are interested mainly in the impact of *imperfect cancellation of interfering signal* in HIC decoder, and hence we compare its performance with two reference successive decoders of data from source C. The reference decoders differ in the amount of *prior knowledge* about $\mathbf{c}_A$, $\mathbf{c}_B$, i.e. in the capability to remove the interfering signal $h_A s_A(c_A) - h_B s_B(c_B)$ from the relay observation (6.1) prior to the decoding of source C data.

"Perfect IC" decoder (P-IC) knows both individual user codewords $\mathbf{c}_A$, $\mathbf{c}_B$ prior to the decoding of $\mathbf{c}_C$, and hence it is capable to perfectly remove the interfering signal $h_A s_A(c_A) - h_B s_B(c_B)$ from its observation (6.1). Contrary to this, *"No IC"* (N-IC) decoder decodes $\mathbf{c}_C$ directly from the relay observation (6.1) and hence the interfering signal is simply

	P-IC	HIC	N-IC
1st decoding	$\mathbf{c}_A$, $\mathbf{c}_B$	$\mathbf{c}_{AB}$	$\mathbf{c}_C$
2nd decoding	$\mathbf{c}_C$	$\mathbf{c}_C$	$\mathbf{c}_{AB}$

Table 6.1: Successive decoders.

ignored as an additional noise. Successive processing of source information in all the analysed decoders is summarised in Table 6.1.

For the sake of simplicity we focus only on the decoding of $\mathbf{c}_C$ at the relay, and hence we assume that the hierarchical codeword $\mathbf{c}_{AB}$ (respectively individual user codewords $\mathbf{c}_A$, $\mathbf{c}_B$) in HIC decoder (respectively P-IC decoder) is provided by the *genie-aided decoder*[5]. In the rest of this section we evaluate the achievable capacity (maximal mutual information for BPSK alphabets) of source C transmission (C_C) in both genie-aided successive decoders of $\mathbf{c}_C$ with interference cancellation (P-IC, HIC) and the interference-blind single-user decoder of $\mathbf{c}_C$ (N-IC). An upper bound of transmission rates from other sources (C_B, C_B, C_{AB}) will be also evaluated for corresponding decoders to provide a relevant comparison with practical (not genie-aided) decoders.

Perfect IC decoder. The capacity $C_C^{\mathrm{P-IC}}$ for the case where both individual user codewords $\mathbf{c}_A$, $\mathbf{c}_B$ are available at the relay prior to decoding of $\mathbf{c}_C$ $(\mathbf{d}_C)$ can be evaluated as:

$$C_C^{\mathrm{P-IC}} = I\left(c_C; y_R | c_A, c_B\right) = \mathcal{H}\left[y_R | c_A, c_B\right] - \mathcal{H}\left[y_R | c_C, c_A, c_B\right]. \tag{6.5}$$

In a practical decoder, both individual user codewords $\mathbf{c}_A$, $\mathbf{c}_B$ has to be decoded from the relay observation (6.1) before the actual decoding of $\mathbf{c}_C$ takes place, which consequently limits the corresponding capacities $C_A^{\mathrm{P-IC}}$, $C_B^{\mathrm{P-IC}}$ of individual single-user decoders. It can be shown that these capacities are equal in a symmetric case:

$$C_i^{\mathrm{P-IC}} = \frac{1}{2} I\left(c_A, c_B; y_R\right) = \frac{1}{2}\left(\mathcal{H}\left[y_R\right] - \mathcal{H}\left[y_R | c_A, c_B\right]\right) \tag{6.6}$$

for $i \in \{A, B\}$, while generally any rate pair inside the conventional MAC capacity region [32] can be achieved in the asymmetric case.

HIC decoder. The capacity C_C^{HIC} for the case where the hierarchical user codeword $\mathbf{c}_{AB}$ is available at the relay prior to decoding of $\mathbf{c}_C$ $(\mathbf{d}_C)$ can be evaluated as the following conditioned mutual information:

$$C_C^{\mathrm{HIC}} = I\left(c_C; y_R | c_{AB}\right) = \mathcal{H}\left[y_R | c_{AB}\right] - \mathcal{H}\left[y_R | c_C, c_{AB}\right]. \tag{6.7}$$

[5] This assumption is equivalent to the case where the transmission rates from sources A, B are limited below the corresponding interference channel capacities (more details will be given later). Moreover, the perfect availability of $\mathbf{c}_{AB}$ (eventually $\mathbf{c}_A$, $\mathbf{c}_B$) represents also the case, where the particular data are received on an independent path from the network prior to the actual decoding of $\mathbf{c}_C$ at the relay.

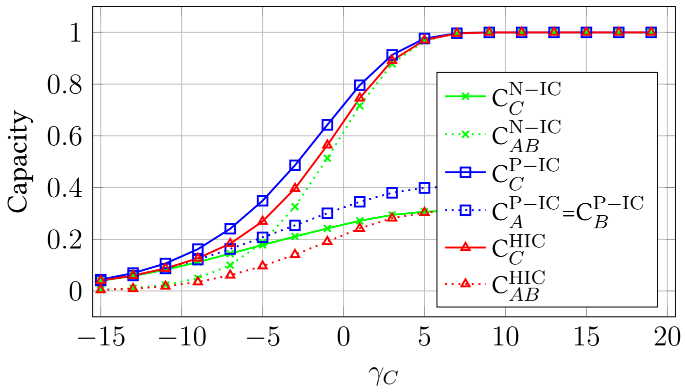


Figure 6.7: Comparison of achievable capacities in P-IC, N-IC and HIC decoders ($h_A = h_B = h_C = 1$; $\gamma_C = \gamma_A = \gamma_B$).

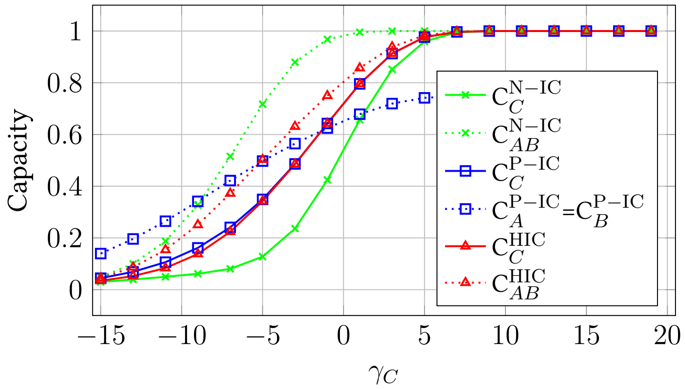


Figure 6.8: Comparison of achievable capacities in P-IC, N-IC and HIC decoders ($h_A = h_B = 1$; $h_C = 1/2$; $\gamma_C = \frac{1}{4}\gamma_A = \frac{1}{4}\gamma_B$).

The requirement to obtain a perfect estimate of $\mathbf{c}_{AB}$ limits the corresponding hierarchical capacity C_{AB}^{HIC} in a practical decoder as:

$$C_{AB}^{\mathrm{HIC}} = I\left(c_{AB}; y_R\right) = \mathcal{H}\left[y_R\right] - \mathcal{H}\left[y_R | c_{AB}\right]. \tag{6.8}$$

No IC decoder. The capacity $C_C^{\mathrm{N-IC}}$ for the case where the relay decoder does not have any knowledge about $\mathbf{c}_A$, $\mathbf{c}_B$, and hence $\mathbf{c}_C$ $(\mathbf{d}_C)$ is decoded directly from the relay observation (6.3) is given by:

$$C_C^{\mathrm{N-IC}} = I\left(c_C; y_R\right) = \mathcal{H}\left[y_R\right] - \mathcal{H}\left[y_R | c_C\right]. \tag{6.9}$$

The N-IC decoder is equivalent to a practical decoder since it does not require any a priori knowledge about other user signals. It allows a perfect IC of signal s_C from relay observation (6.2) and subsequent decoding of hierarchical codeword $\mathbf{c}_{AB}$ from the resulting interference-free channel. Consequently, the capacity of $C_{AB}^{\mathrm{N-IC}}$ is given directly by the hierarchical channel capacity, see [1]:

$$C_{AB}^{\mathrm{N-IC}} = I\left(c_{AB}; y_R | c_C\right) = \mathcal{H}\left[y_R | c_C\right] - \mathcal{H}\left[y_R | c_C, c_{AB}\right]. \tag{6.10}$$

6.1.5 Numerical evaluation

All alphabet constrained capacities were evaluated numerically by the Monte-Carlo integral evaluation (for details see [1] and references therein). SNR of source $i \to$ relay channel ($i \in \{A, B, C\}$) is defined as $\gamma_i = \mathrm{E}\left[{|h_i|^2 |s_i|^2}/{\sigma_w^2}\right]$.

A comparison of capacities $C_C^{\mathrm{P-IC}}$, C_C^{HIC} and $C_C^{\mathrm{N-IC}}$ is provided in Figures 6.7 and 6.8 for two different examples of channel parameterisation. All capacities are evaluated as a function of SNR on source $C \to$ relay channel (γ_C). As expected, the performance of *hierarchical interference canceller* (HIC) is bounded by the performance of *perfect interference canceller* (P-IC) and *interference blind* receiver (N-IC) as:

$$C_C^{\mathrm{N-IC}} \leq C_C^{\mathrm{HIC}} \leq C_C^{\mathrm{P-IC}}. \tag{6.11}$$

As it is also evident from Figures 6.7 and 6.8 the actual performance of all successive decoders (see Table 6.1) is sensitive to actual channel parameterisation. To analyse the influence of channel parameterisation on the performance, we evaluate also the mean achievable capacities $\bar{C} = \mathrm{E}_{\varphi}\left[C\left(\varphi\right)\right]$, where $\varphi = [\angle h_A; \angle h_B; \angle h_C]$ and $\angle h_i$ is uniformly distributed on $\langle 0; 2\pi)$, $i \in \{A, B, C\}$. The comparison of mean capacities $\bar{C}_C^{\mathrm{P-IC}}$, $\bar{C}_C^{\mathrm{HIC}}$ and $\bar{C}_C^{\mathrm{N-IC}}$ is provided in Figures 6.9 and 6.10.

As it is obvious from a comparison of Figures 6.7 and 6.8 with Figures 6.9 and 6.10, the HIC decoder performance suffers only from a *minor SNR loss* when compared with the upper bound given by the P-IC decoder (i.e. $C_C^{\mathrm{HIC}} \approx C_C^{\mathrm{P-IC}}$). This observation *justifies a viability of hierarchical interference cancellation* in WPLNC decoders. However, it is important to note again, that the maximal capacities of source C transmission in HIC (respectively P-IC) decoders are conditioned by the availability of perfect estimate of $\mathbf{c}_{AB}$ (respectively $\mathbf{c}_A$, $\mathbf{c}_B$) prior to the decoding of source C data. In the practical receiver (not genie-aided) this assumption implicitly forces an upper bound of transmission rates from sources A, B. To make the picture complete, these limits on C_{AB}^{HIC} (HIC decoder) or $C_A^{\mathrm{P-IC}}$, $C_B^{\mathrm{P-IC}}$ (P-IC decoder) are evaluated in Figures 6.7, 6.8, 6.9 and 6.10 as well[6].

[6] In the N-IC decoder, to the contrary, the capacity $C_{AB}^{\mathrm{N-IC}}$ represents only the maximal transmission rate of hierarchical codewords, which can be achieved after $\mathbf{c}_C$ is decoded (see the order of successive decoding operations in Table 6.1).

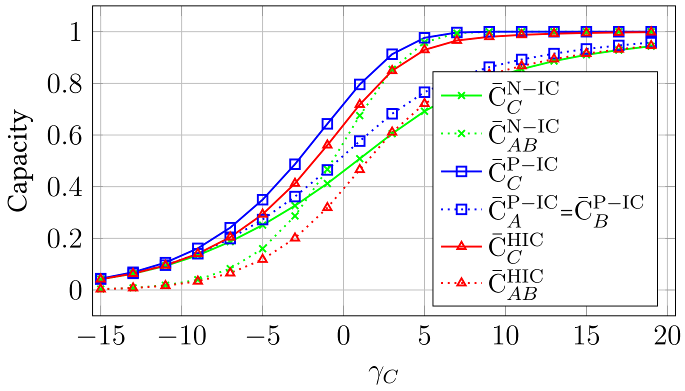


Figure 6.9: Comparison of mean achievable capacities in P-IC, N-IC and HIC decoders ($\angle h_i \in \langle 0; 2\pi)$, $|h_i| = 1$, $i \in \{A, B, C\}$; $\gamma_C = \gamma_A = \gamma_B$).

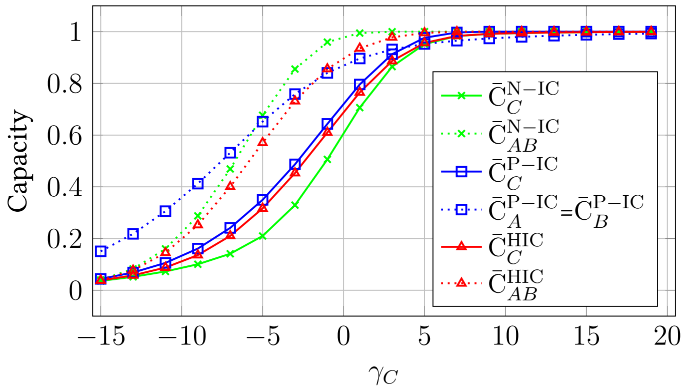


Figure 6.10: Comparison of mean achievable capacities in P-IC, N-IC and HIC decoders ($\angle h_i \in \langle 0; 2\pi)$, $i \in \{A, B, C\}$, $|h_A| = |h_B| = 1$, $|h_C| = 1/2$; $\gamma_C = \frac{1}{4}\gamma_A = \frac{1}{4}\gamma_B$).

6.1.6 Conclusion

We have shown that hierarchical interference cancellation is a viable decoding strategy in WPLNC systems, since it suffers only from some *residual interference*, due to an imperfect removal of interfering hierarchical signal. We have observed only a minor performance degradation (when compared to perfect interference cancellation) of the SD-HIC decoding strategy in the binary-modulated 3-source relay channel. In the future work we would like to analyse the achievable capacity regions of multi-user WPLNC system and identify the optimal decoding strategies to respect all constraints induced by practical (not genie aided) decoders.

6.2 Joint and recursive HIC with successive CaF decoding

6.2.1 Introduction

Background and related work

WPLNC is the network structure aware PHY layer coding technique currently receiving a wide attention in the research community. The WPLNC technique utilises the knowledge of the network structure directly at the PHY layer to improve the efficiency of the communication in a complex network. To some extent it reduces the need of separate Medium Access Control or Network Routing Layer. The information is "flooded" through the whole network directly at PHY layer coding and processing in a form of many-to-one functions (Hierarchical Network Code maps (HNC map)) of data. This reduces the size of (hierarchical) codebook that needs to be decoded and also the rates transmitted from the relay while fully respecting all PHY constellation space related aspects – mainly the parameterisation of channels. The final destination collects noisy observation of several HNC maps and solves (jointly decodes) for the desired source data. The WPLNC can be also seen as the step from the network coding principles [37] designed for discrete finite field alphabets over lossless links to the signal space representation typical for wireless communications.

There are many possibilities of designing Network Coded Modulation (NCM) and related relay processing (see [79], [21] for an overview). The NCM can be based on traditional single user codes and traditional constellations with layered approach properly addressing exclusive law (solvability) of the HNC maps at constellation level and relays using Hierarchical Decode & Forward strategy [1]. Selected results are available investigating various particular forms of the source, relay and destination strategies: lattice based [19], [118], [15] and other [13] network aware multi-source encoding, adaptive relay mapping/decision strategies [12], [9], [12], [11].

CaF technique [22] is a generalisation of lattice based design approach. It uses nested lattices and receiver modulo lattice operation complemented with Minimum Mean Square

Error (MMSE) equaliser alignment of lattices misaligned by the channel fading coefficients. This performs a constellation space equivalent operation to HNC map function on discrete data themselves. This section focuses on CaF strategy.

Goals of this section and contributions

The motivation and goals of this section can be outlined in the following points.

1. CaF relay processing technique complemented with nested lattice type of NCM is believed to be "all-in-one" solution for designing WPLNC based networks. However we see that it has several deficiencies. The major one is the fact that receiver lattice mismatch is aligned only through the *scalar single tap equaliser*. It does not provide enough degrees of freedom for multiple misaligned sources.

2. Different HNC decodable (with complying computation rate) maps represent various forms of structure knowledge that should be exploited at the receiver. These maps are however still many-to-one functions having ambiguity in their constellation space (lattice) representation.

3. The standard CaF assumes essentially that the number of involved source nodes and relays is high and we have plenty of choices to optimise HNC map coefficients that maximise the computation rate. However in practical situations with a small number of nodes, we are rather limited and frequently only few are allowed in order to guarantee final destination solvability. Respecting subsequent stages of the network (not just the MAC (Multiple Access Channel) stage of the first stage) frequently dictates further constraints on the map that is required to be processed by the relay. A *particular* HNC map is typically *desired* to be processed by the relay.

4. The goal of this work is *extension of CaF* processing which allows to *increase degrees of freedom* in processing (equalising) misaligned lattices and to *utilise all available hierarchical codebook* (HNC map) structure knowledge.

In this section we provide the following contributions and results.

1. We introduce a concept of Hierarchical Interference Cancellation (H-IFC). It describes the technique where we help the decoder of given *desired* HNC map by partially mitigating the ambiguity generated by all involved sources by using other (again many-to-one) HNC map structure knowledge. Number of known maps is assumed not to be sufficient for fully resolving all contributing sources (thus the name "hierarchical" IFC).

2. The H-IFC concept will be particularly used in extending and generalising CaF technique. It will use Hierarchical Successive CaF Decoding (SCFD) where multiple auxiliary many-to-one HNC maps are decoded. They are subsequently used

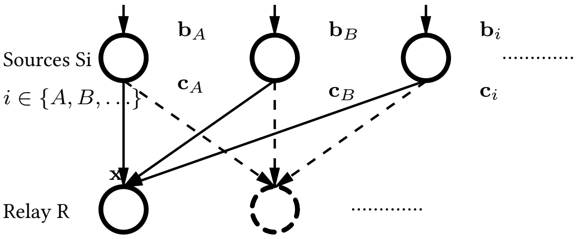


Figure 6.11: Hierarchical MAC single stage network fragment.

in H-IFC where each auxiliary map increases the degrees of freedom of equaliser aligning lattices misaligned by channel fading coefficients.

3. This technique will be presented in two variants. The first performs H-IFC equalisation jointly for all auxiliary maps at once. The second form does that in recursive manner, each map at one step.

6.2.2 Preliminaries

System model and definitions

We assume a Hierarchical MAC single stage fragment of the WPLNC network (Figure 6.11). This fragment can be used as a building block of a complex network. For the notational simplicity, we will assume only a single relay R. Generalisation for multiple relays is straightforward. Sources Si, $i \in \{A, B, \ldots\}$ with data vectors $\mathbf{b}_i$ transmit codewords (nested lattice points) ($\mathbf{c}_i \in \mathbb{C}^n$) $\mathbf{c}_i = \mathbf{c}_i(\mathbf{b}_i)$. Channel between source Si and the relay is a block flat fading AWGN channel with fading coefficient $h_i \in \mathbb{C}$ and noise $\mathbf{w}$ with σ_w^2 variance per dimension. The received signal is

$$\mathbf{x} = \sum_i h_i \mathbf{c}_i + \mathbf{w}. \tag{6.12}$$

Compute-and-forward relay strategy

Here we very briefly summarise CaF strategy for readers convenience and for notation definition purposes (see [22] for details). CaF is a particular form of NCM based on nested lattice codes where the relay can decode a *linear* function of the data vectors from involved sources. It uses a linear single tap equaliser which minimises (in MMSE sense) the impact of the channel parameterisation. Essentially, it scales the superposed lattices misaligned by channel coefficients to “fit” as good as possible to the common lattice. Then it is followed by the modulo shaping lattice operation which equivalently corresponds to Gallois Field (GF) linear function on source data symbols.

Sources transmit Λ_S-shaped Λ_C-lattice codewords $\mathbf{c}_i = \mathbf{c}_i(\mathbf{b}_i)$ with $P_i = P$ power and rates $R_i, R_{i'} \leq R_A$ (lower rate codes are zero-padded). Individual data symbols are from GF $b_i \in \mathbb{F}_M$. The relay has the observation (6.12).

The relay decodes linear minimal cardinality [79] HNC map $\mathbf{b} = \sum_i q_i \mathbf{b}_i$, $q_i \in \mathbb{F}_M$. Modulo-equivalent codeword HNC maps are formed by coefficients $a_i \in \mathbb{Z}$, where

$$\mathbf{a} = [a_A, a_B, \ldots]^T$$

, such that

$$\sum_i a_i \mathbf{c}_i \bmod \Lambda_S = \mathbf{c}(\mathbf{b}). \tag{6.13}$$

Coefficients $\mathbf{a}$ are such that after $\mathrm{mod}\Lambda_S$ lattice preprocessing we get the valid codeword $\mathbf{c}(\mathbf{b})$. Coefficients can be back-mapped on data symbol combination coefficients $a_i \bmod M \mapsto q_i$.

Before applying modulo shaping lattice operation, the receiver performs *linear scalar equalisation*

$$\tilde{\mathbf{x}} = \alpha \mathbf{x} = \alpha \sum_i h_i \mathbf{c}_i + \alpha \mathbf{w} \tag{6.14}$$

with MMSE “matching” of codeword HNC map misalignment $\rho = \|\alpha \mathbf{x} - \sum_i a_i \mathbf{c}_i\|^2$

$$\hat{\alpha} = \arg\min_{\alpha} \mathrm{E}\left[\| \sum_i (\alpha h_i - a_i)\mathbf{c}_i + \alpha \mathbf{w}\|^2 \right]. \tag{6.15}$$

A close-form solution is

$$\hat{\alpha} = \frac{P\mathbf{h}^H \mathbf{a}}{\sigma_w^2 + P\|\mathbf{h}\|^2}. \tag{6.16}$$

The residual lattice misalignment interference mean power is $P\|\hat{\alpha}\mathbf{h} - \mathbf{a}\|^2$.

Computation rate at the relay R is then

$$\tilde{R}(\mathbf{a}) = \max \lg^+ \left(\frac{P}{|\alpha|^2 \sigma_w^2 + P\|\alpha \mathbf{h} - \mathbf{a}\|^2} \right) \tag{6.17}$$

where $\lg^+ x = \max(0, \log_2 x)$. It is maximised by MMSE equaliser with coefficient $\hat{\alpha}$ given by (6.16) and is achievable by nested lattice code [22]. Computation rate $\tilde{R}(\mathbf{a})$ determines maximum hierarchical rate at the relay. All sources with $a_i \neq 0$ have to have rates $R_i < \tilde{R}(\mathbf{a})$.

Multiple HNC map CaF

It might happen that our *desired* map $\mathbf{a}$ does not have a sufficient computation rate. Instead, the relay can compute multiple (auxiliary) maps $\{\tilde{\mathbf{a}}^m\}_m$ provided that their computation rate is higher than the rate of all participating sources and that the desired map can be get as a *linear integer* combination

$$\mathbf{a} = \sum_m \eta_m \tilde{\mathbf{a}}^m. \tag{6.18}$$

In [22], it is called Successive Cancellation. A similar idea also appears in [117]. Here, all interference is assumed to have a common fading coefficient and then, 2 standard CaF maps are used to solve desired and interfering lattices. There is also described a variant which sequentially nulls interference with regard to desired lattice.

All these ideas share a common principle of computing several auxiliary maps (with potentially more favorable computation rate) instead of the desired one and then obtain the desired one as a linear integer combination. We will use this as a *reference* case and we call it *multiple map CaF.*

Hierarchical interference cancelation

In this section, we present two variants of CaF generalisation which introduce additional degrees of freedom into the lattice misalignment equaliser and allows exploiting all available structure knowledge of hierarchical codewords (many-to-one source codeword functions). An additional advantage is that the auxiliary and the desired map does not need to be linearly dependent. Two variants will be introduced: with joint and recursive processing.

H-SCFD with joint H-IFC multi-tap equaliser

Hierarchical Successive CaF Decoding (H-SCFD) with *Joint* H-IFC multi-tap equaliser performs the following steps.

1. **H-SCFD step:** The relay decodes multiple auxiliary IFC HNC maps $\{\tilde{\mathbf{a}}^m\}_m$. Each map has to have sufficient computation rate for all participating sources $R_i < \tilde{R}(\mathbf{a})$ for all i such that $a_i \neq 0$.
2. **Joint H-IFC equaliser step:** Let the desired map be $\mathbf{a}$ (it does not need to be a linear combination of auxiliary maps). Joint H-IFC multi-tap equaliser minimises MSE lattice misalignment with regard to to desired map $\mathbf{a}$ including multi-tap scaling of all available decodable hierarchical codeword structures

$$[\hat{\alpha}, \hat{\boldsymbol{\beta}}] = \arg\min_{\alpha,\boldsymbol{\beta}} \mathrm{E}\left[\|\alpha \mathbf{x} - \sum_m \beta_m \left(\sum_i \tilde{a}_i^m \mathbf{c}_i \right) - \sum_i a_i \mathbf{c}_i \|^2 \right] \tag{6.19}$$

where $\boldsymbol{\beta} = [\ldots, b_m, \ldots]$.

Notice, that all exploitable decodable hierarchical codewords map have dedicated interface canceller equaliser tap which is optimised for this map. This means that we exactly as many equaliser degrees of freedom as is the number of exploitable patterns. It is worth stressing that these patterns are only hierarchical, i.e. many-to-one functions of the source codeword and as such they still have constellation/lattice space ambiguity. This is the reason we use "hierarchical" IFC name. Please compare that to classical successive decoding interference canceller where an exact and non-ambiguous form of source codeword is subtracted.

Next we find a closed form solution of the MMSE Joint H-IFC equaliser. For a simplicity, we assume only a single IFC map $\tilde{\mathbf{a}}$. Generalisation is straightforward. Single auxiliary map Joint H-IFC MMSE equaliser is stated as

$$[\hat{\alpha}, \hat{\beta}] = \arg\min_{\alpha,\beta} \mathrm{E}\left[\|\alpha\mathbf{x} - \beta \sum_i \tilde{a}_i \mathbf{c}_i - \sum_i a_i \mathbf{c}_i\|^2 \right]. \tag{6.20}$$

The utility function is

$$\rho = \mathrm{E}\left[\|\alpha\mathbf{x} - \beta \sum_i \tilde{a}_i \mathbf{c}_i - \sum_i a_i \mathbf{c}_i\|^2 \right] \tag{6.21}$$

$$= nP\|\alpha\mathbf{h} - \beta\tilde{\mathbf{a}} - \mathbf{a}\|^2 + n|\alpha|^2\sigma_w^2 \tag{6.22}$$

is a real-valued function of complex variables α, β. In order to find its stationary point, we need to use *generalised* derivative, see [119]. Performing this leads to linear equations easily solvable in a closed form. MMSE coefficients are get from

$$\begin{bmatrix} \gamma\|\mathbf{h}\|^2 + 1 & -\gamma\mathbf{h}^H\tilde{\mathbf{a}} \\ \gamma\tilde{\mathbf{a}}^H\mathbf{h} & -\gamma\|\tilde{\mathbf{a}}\|^2 \end{bmatrix} \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} \gamma\mathbf{h}^H\mathbf{a} \\ \gamma\tilde{\mathbf{a}}\mathbf{a} \end{bmatrix} \tag{6.23}$$

where $\gamma = P/\sigma_w^2$ is Signal-to-Noise Ratio (SNR).

Residual IFC misalignment is $z = \sum_i \left(\hat{\alpha}h_i - \hat{\beta}\tilde{a}_i - a_i\right)\mathbf{c}_i$ with mean power

$$P_z = \frac{\mathrm{E}\left[\|z\|^2\right]}{n} = P\|\hat{\alpha}\mathbf{h} - \hat{\beta}\tilde{\mathbf{a}} - \mathbf{a}\|^2. \tag{6.24}$$

Resulting computation rate is

$$\tilde{R}_J = \mathrm{lg}^+ \frac{P}{|\hat{\alpha}|^2\sigma_w^2 + P\|\hat{\alpha}\mathbf{h} - \hat{\beta}\tilde{\mathbf{a}} - \mathbf{a}\|^2}. \tag{6.25}$$

H-SCFD with recursive H-IFC equaliser

Hierarchical Successive CaF Decoding (H-SCFD) with *Recursive* H-IFC equaliser performs recursively equaliser steps for each auxiliary map at one step.

1. **H-SCFD step:** The relay decodes multiple auxiliary IFC HNC maps $\{\tilde{\mathbf{a}}^m\}_m$. Each map has to have sufficient computation rate for all participating sources $R_i < \tilde{R}(\mathbf{a})$ for all i such that $a_i \neq 0$.

2. **H-IFC equaliser at recursion m step:** Let the desired map be $\mathbf{a}$ (it does not need to be a linear combination of auxiliary maps). At each recursion step, the equaliser minimises MSE lattice misalignment with regard to to desired map $\mathbf{a}$ using only scaling of m-th decodeable structure (map $\tilde{\mathbf{a}}^m$)

$$\hat{\beta}_m = \arg\min_{\beta_m} \mathrm{E}\left[\|\mathbf{y}_{m-1} - \beta_m \left(\sum_i \tilde{a}_i^m \mathbf{c}_i \right) - \sum_i a_i \mathbf{c}_i\|^2 \right] \tag{6.26}$$

where we define the initialisation $\mathbf{y}_0 = \mathbf{x}$. The H-IFC canceller is then

$$\mathbf{y}_m = \mathbf{y}_{m-1} - \hat{\beta}_m \sum_i \tilde{a}_i^m \mathbf{c}_i. \tag{6.27}$$

3. **Final CaF equaliser step:** At the end, for *final* $\mathbf{y}$ we perform a standard CaF α MMSE equalisation (see Section 6.2.2). Effectively it means the standard CaF is applied to new effective channel coefficients

$$\tilde{\mathbf{h}} = \mathbf{h} - \sum_m \hat{\beta}_m \tilde{\mathbf{a}}^m. \tag{6.28}$$

A closed-form solution will be for a simplicity again show for single auxiliary map $\tilde{\mathbf{a}}$ (with a straightforward generalisation). Optimal MMSE H-IFC coefficient solution is again get using generalised derivative of the utility function

$$\rho = \mathrm{E}\,\|\mathbf{x} - \beta \sum_i \tilde{a}_i \mathbf{c}_i - \sum_i a_i \mathbf{c}_i\|^2 \tag{6.29}$$

$$= nP\|\mathbf{h} - \beta\tilde{\mathbf{a}} - \mathbf{a}\|^2 + n\sigma_w^2. \tag{6.30}$$

The solution has a form of a projector

$$\hat{\beta} = \frac{\tilde{\mathbf{a}}^H(\mathbf{h} - \mathbf{a})}{\|\tilde{\mathbf{a}}\|^2} \tag{6.31}$$

and does not depend on SNR.

The resulting new effective channel is then $\tilde{\mathbf{h}} = \mathbf{h} - \hat{\beta}\tilde{\mathbf{a}}$. This effective channel can be easily plugged into the standard CaF solution. The computation rate is

$$\tilde{R}_1(\mathbf{a}_j) = k \lg^+ \frac{P}{|\hat{\alpha}|^2 \sigma_w^2 + P\|\hat{\alpha}\tilde{\mathbf{h}} - \mathbf{a}\|^2} \tag{6.32}$$

with optimal MMSE α (CaF for the effective channel)

$$\hat{\alpha} = \frac{P\tilde{\mathbf{h}}^H \mathbf{a}}{\sigma_w^2 + P\|\tilde{\mathbf{h}}\|^2}. \tag{6.33}$$

6.2.3 Numerical results

We examined both suggested one-step variants of hierarchical interference canceller in the 3-user MAC. Standard CaF with multiple HNC maps was used as a reference scenario. In the first case, we considered channel coefficients causing only *small* lattice misalignment. Thus, we chose real valued vector of channel coefficients $\mathbf{h} = [3.1, 2.1, 0.9]$. Without loss of generality we assumed desired HNC map to be $\mathbf{a} = [1, 1, 1]$. In the reference case, we chose HNC map $\tilde{\mathbf{a}}^1 = \mathbf{a}_h = [1, 1, 0]$ maximising the computation rate at SNR = 5 dB and second auxiliary map $\tilde{\mathbf{a}}^2 = \mathbf{a}_x = [0, 0, -1]$. These maps were selected in order to obtain the desired map $\mathbf{a}$ as their integer linear combination. For recursive and joint H-SD-IFC auxiliary HNC maps, the map maximising the computation rate of the desired map was used. For a given $\mathbf{h}$ we obtained $\tilde{\mathbf{a}} = \mathbf{a}_{\text{ifc}} = [-2, -1, 0]$. In Figure 6.12a, there are plotted computation rates of individual maps. It is obvious that the resulting achievable rate of appropriate scenario is given by the minimal rate among all participating HNC maps. Achievable rates for all considered scenarios are plotted in Figure 6.12b, where we can see that standard CaF with multiple maps is not optimal for given $\mathbf{h}$, while both hierarchical interference cancellers gave the computation rates above the case of only one desired HNC map.

The same analysis was performed for channel coefficients causing *large* lattice misalignment. An example of such channel is $\mathbf{h} = [3.3, 1.5, 0.9]$. Appropriate HNC maps together with their computation rates are shown in Figure 6.13a. From results in Figure 6.13b it can be seen that standard CaF with chosen multiple maps has poor performance again. Recursive H-SD-IFC gave better results, even without restriction on desired map to be a linear integer combination of auxiliary HNC maps.

In [22, 117] there is mentioned that CaF extension to complex valued channel is straightforward, but all papers about CaF give numerical results only for a real valued case. Therefore, we also show the results of considered scenarios for complex channel coefficients (Figure 6.14 and Figure 6.15). Again we can see that recursive H-SD-IFC ranks among the scenarios with better performance. In Figure 6.14 there is shown that CaF with multiple auxiliary maps can also achieve good results, even better than joint H-SD-IFC.

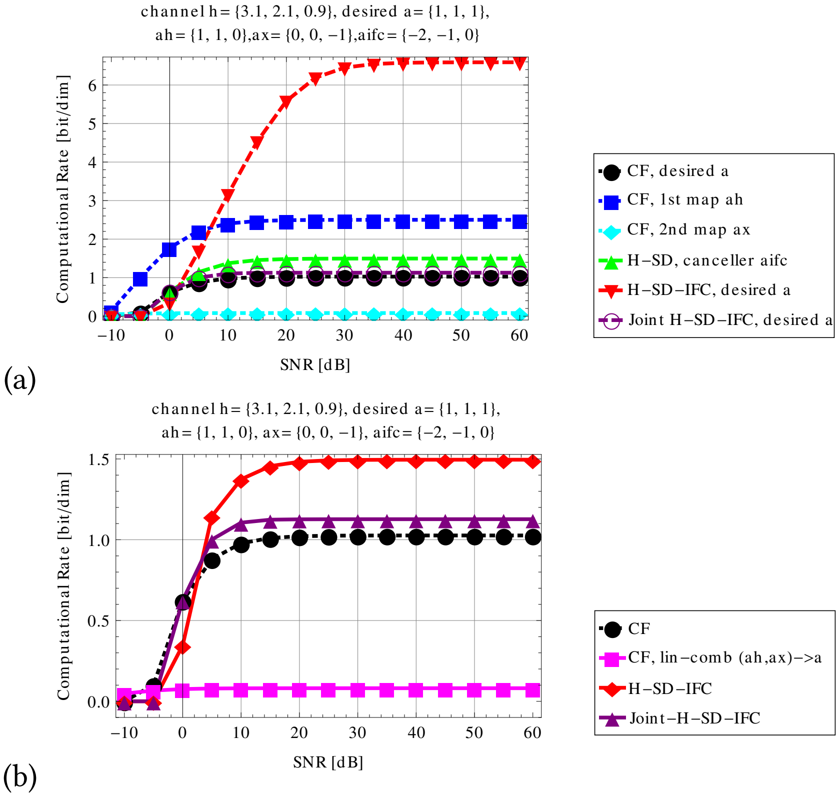
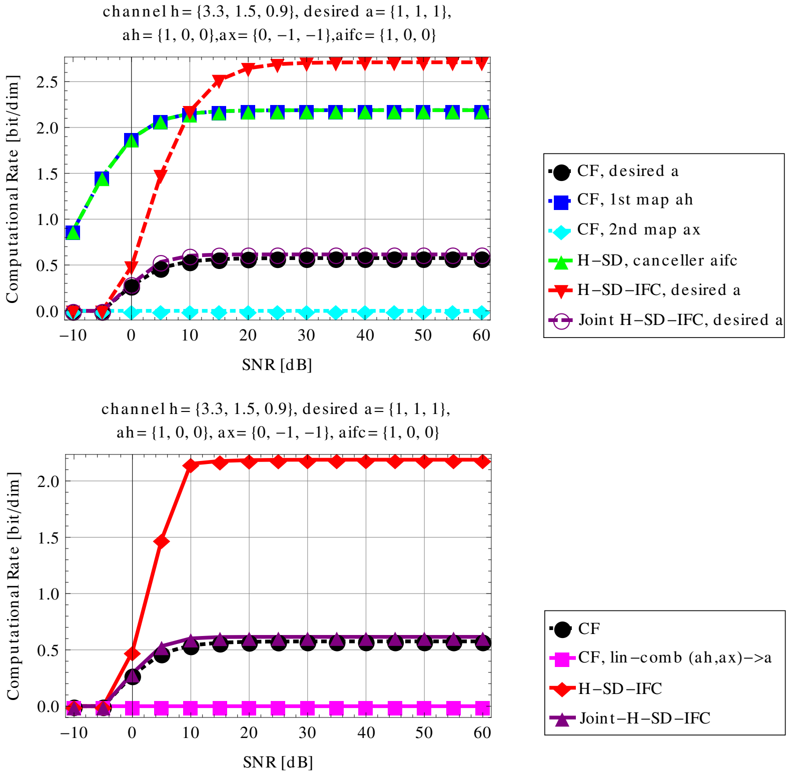


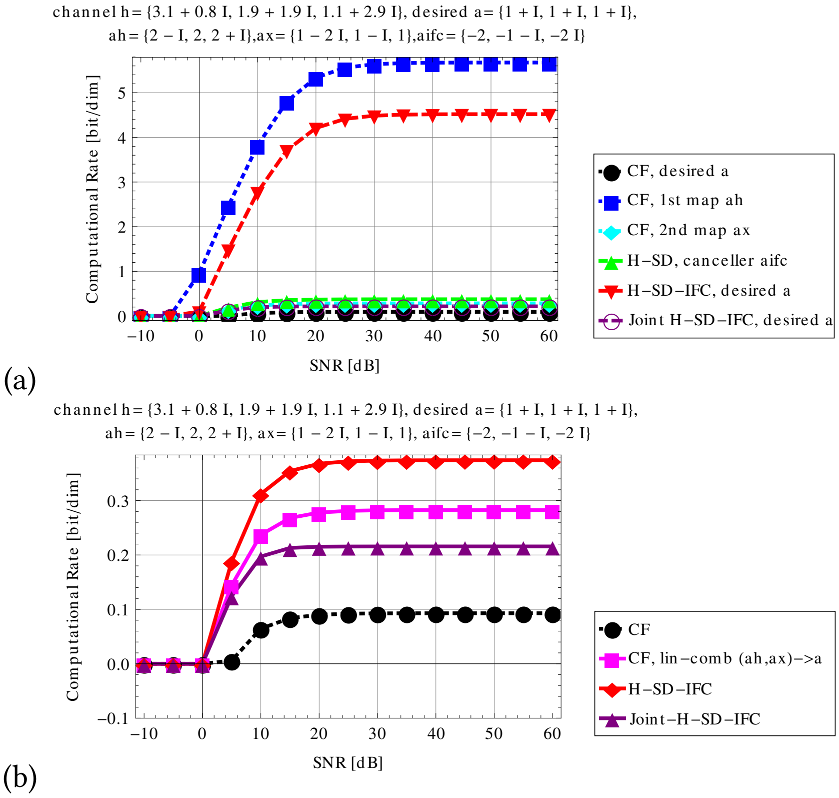
Figure 6.12: Real valued channel with small lattice misalignment: (a) Computation rates for all participating auxiliary HNC maps, (b) Resulting computation rates as minimum over all auxiliary HNC maps associated with given scenario.



(a)

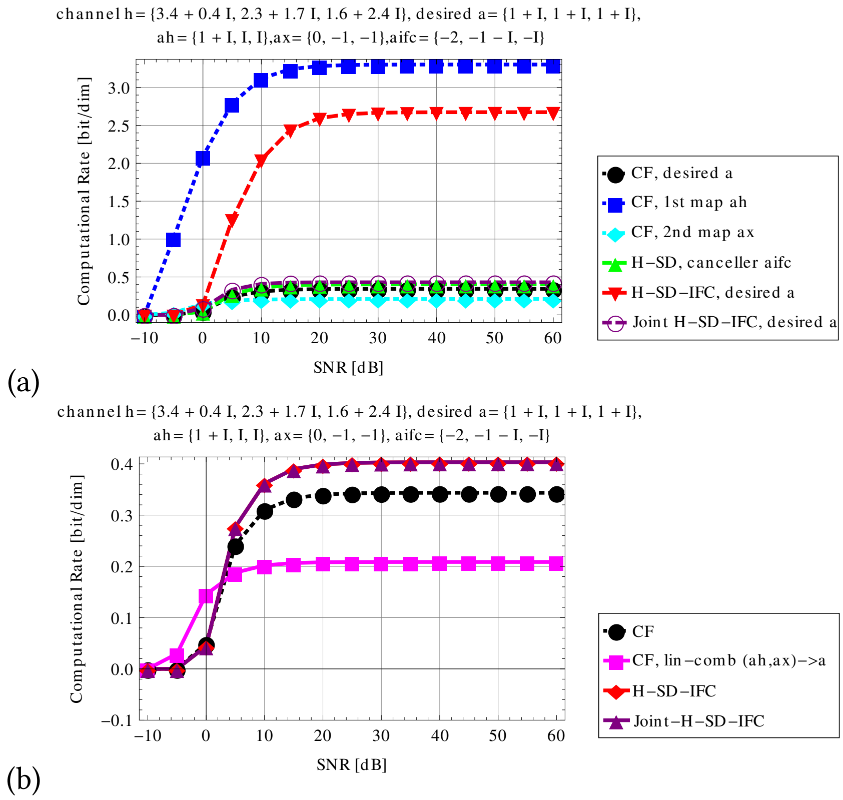
(b)

Figure 6.13: Real valued channel with large lattice misalignment: (a) Computation rates for all participating auxiliary HNC maps, (b) Resulting computation rates as minimum over all auxiliary HNC maps associated with given scenario.


channel h= {3.1 + 0.8 I, 1.9 + 1.9 I, 1.1 + 2.9 I}, desired a= {1 + I, 1 + I, 1 + I}, ah= {2 − I, 2, 2 + I},ax= {1 − 2 I, 1 − I, 1},aifc= {−2, −1 − I, −2 I}

channel h= {3.1 + 0.8 I, 1.9 + 1.9 I, 1.1 + 2.9 I}, desired a= {1 + I, 1 + I, 1 + I}, ah= {2 − I, 2, 2 + I}, ax= {1 − 2 I, 1 − I, 1}, aifc= {−2, −1 − I, −2 I}

Figure 6.14: Complex valued channel with small lattice misalignment: (a) Computation rates for all participating auxiliary HNC maps, (b) Resulting computation rates as minimum over all auxiliary HNC maps associated with given scenario.



(a)

(b)

Figure 6.15: Complex valued channel with large lattice misalignment: (a) Computation rates for all participating auxiliary HNC maps, (b) Resulting computation rates as minimum over all auxiliary HNC maps associated with given scenario.

6.2.4 Discussion and conclusions

We developed a generalisation of CaF relay strategy that increases number of degrees of freedom in lattice misalignment equaliser using hierarchical CaF successive decoding. Apart of creating multi-tap equaliser we also utilise all available auxiliary HNC map codebook knowledge in relay CaF processing. Freedom gained in this multi-tap equaliser-canceller clearly allows achieving better computation rates and also relaxes the need of *linear only* HNC map combination in standard multi-map CaF. In a complicated system, we can set the desired map paying less attention to its optimality with regard to CaF misalignment equalisation.

Numerical results showed that our hierarchical interference canceller can achieve good results, even for channel coefficients causing large lattice misalignment. However, for each channel coefficient vector, we can find at least one HNC map leading to catastrophic degradation of computation rate. Nevertheless, HNC map with poor performance in one scenario can be better for another one. Therefore, some adaptive mixture of H-IFC strategies can lead to desired performance.

7 Performance analysis

7.1 Hierarchical pairwise error probability for hierarchical decode and forward strategy in WPLNC

7.1.1 Introduction

Pairwise Error Probability (PEP) is an essential tool allowing to design practical optimised coding schemes. It reveals the connection between the performance and the decoding metric that is directly related to the codeword and/or constellation properties. The application of this principle to the *hierarchical* PEP used to describe the decoding performance of *many-to-one* message functions (Hierarchical Network Code maps) in WPLNC is addressed in this paper. Unlike for the single-user case, the *hierarchical PEP* reveals a complicated dependence on the structure of the hierarchical codeword/constellation. The structure is defined in terms of *hierarchical distance* and *hierarchical self-distance spectra.* We show that the Network Coded Modulation (NCM) minimising the hierarchical decoding error probability should have zero self-distance spectrum leading to *self-folded NCM design criterion.*

WPLNC background

WPLNC communication networks deliver the information from sources to destinations through the complex network of relays. The information flow does not route individual source messages but their many-to-one functions, similarly to network coding. But in addition to this, the signals of multiple transmitting nodes are allowed to interact at the receiving relay and the relay performs all processing directly in the constellation space. The extraction of the many-to-one message function out of the interacting signals can be done in multiple *hierarchical* encapsulation levels, and we name it *hierarchical* information. Network Coded Modulation (NCM) denotes the set of component constellation space codebooks aware of the network structure and designed for a particular relay strategy. Relay strategies can have many forms: Compute and Forward (CaF), Denoising, Compress and Forward, Hierarchical Decode and Forward (HDF), etc. Final destination node determines the desired message by solving the decoding task on multiple collected received signals carrying hierarchical messages with independent many-to-one message functions. The overview of selected strategies can be found in [21, 22, 75–79], and selected results related to the error rate performance are in [50, 120].

Motivation

The error probability is an important performance indicator and also an obvious code and receiver design optimisation goal. The exact error probability evaluation based on the transition probabilities is too complex (apart of trivial uncoded cases) to be practically useful. It also gives only a limited insight for a *synthesis* of the code. A pairwise error probability can be used to upper-bound the true error rate. As a side-effect it also *connects* the performance target with the metric used by the demodulator and decoder. This can be used for the code synthesis.

This paper focuses on the HDF relay strategy [78] in a single-stage H-MAC channel.[1] The relay wants to decode the H-message. Multiple combinations of component codewords correspond to one H-message. As a consequence, the hierarchical symbol/codeword, is generally *a set of multiple* constellation points or codewords $\mathcal{U}(b)$. It is called called H-constellation.[2] The fact that H-constellation (or H-codeword) has *multiple* representations of the H-message makes the pair-wise error analysis *substantially more difficult.* Also the *interpretation* of the results reveals some surprising facts which do *not* have their equivalents in the classical single user system.

Contribution and main results

We will define hierarchical pairwise error probability and we will show how this can be used in the isomorphic layered NCM design. The evaluation of the hierarchical error probability on the relay implies *H-message relay decoding strategy*, i.e. HDF.

(1) We derive the expression for *Hierarchical Pair-wise Error Probability (H-PEP)* revealing its dependence on the structure of the hierarchical constellation. The structure is defined in terms of *hierarchical distance* and *hierarchical self-distance spectra.*

(2) We show that the NCM minimising the hierarchical decoding error probability should have *zero* self-distance spectrum and we call this case *self-folded NCM.* The self-folding property provides a neat NCM design criterion.

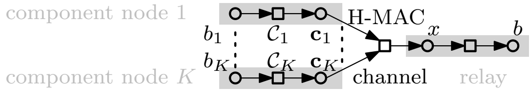


Figure 7.1: System model for H-MAC stage.

7.1.2 System model and definitions

H-MAC stage

In order to streamline the development of the paper, we focus on a generic *single* H-MAC stage with a *single* receiving relay (Figure 7.1). It is a smallest building block of a more complex WPLNC system. The component messages $b_k \in [1 : M]$ of all transmitting nodes are encoded by NCM into $\mathbf{c}_k = \mathcal{C}_k(b_k)$, $k \in [1 : K]$, where K is the number of the transmitters in H-MAC. The relay wants to decode H-message which is Hierarchical Network Code (HNC) map (many-to-one function) $b = \chi(\tilde{b})$ of the component messages, where the concatenation of all component messages is $\tilde{b} = [b_1, \ldots, b_K]$. The "tilde" denotes "concatenation" and will also be used in the notations of codewords $\tilde{\mathbf{c}} = [\mathbf{c}_1^T, \ldots, \mathbf{c}_K^T]^T$, symbols $\tilde{c} = [c_1^T, \ldots, c_K^T]^T$, codebooks $\tilde{\mathcal{C}} = [\mathcal{C}_1, \ldots, \mathcal{C}_K]$. The observation is described by $p(\mathbf{x}|\tilde{b}) = p(\mathbf{x}|\tilde{\mathbf{c}}(\tilde{b}))$, where $\mathbf{x} = \mathbf{u}(\tilde{\mathbf{c}}) + \mathbf{w}$, $\mathbf{u}(\tilde{\mathbf{c}})$ is the channel-combined constellation symbol depending on all component codewords $\tilde{\mathbf{c}}$, and $\mathbf{w}$ is Gaussian noise.

Isomorphic layered NCM

We call the set of component codebooks $\{\mathcal{C}_k\}_{k=1}^K$ *isomorphic layered NCM* if there exists one-to-one mapping H-code $\mathcal{C}$ and H-codeword HNC map $\mathbf{c} = \chi_c(\tilde{\mathbf{c}})$ such that $\mathbf{c} = \mathcal{C}(b)$. The isomorphism implies that the component-wise encodings can be equivalently expressed for the relation between the desired H-message and the corresponding H-codeword. Since the H-decoding can be based on the decoding metric related to the H-codeword which is first obtained from the component-wise channel observation model $p(\mathbf{x}|\tilde{\mathbf{c}})$, we call that approach also layered. The examples of isomorphic layered NCM are CaF [22] or layered block linear NCM [78].

The analysis in this paper can be generally applied on the level of complete messages b, codewords $\mathbf{c}$ and vector observation $\mathbf{x}$ but it could also be applied in the symbol-wise manner (c_n, x_n), e.g. for the uncoded case. We will use a generic notation using b, c, x to cover both.

[1] A prefix "H-" is used to denote "hierarchical" entity, i.e. the one that is generally many-to-one function of the components, or a processing (e.g. codebook, codeword, MAC channel, etc.) related to the hierarchical entities.

[2] A trivial uncoded example has two component BPSK sources with symbols $\{\pm 1\}$. The channel combined constellation points are $\{-2, 0, 2\}$. For XOR type of many-to-one hierarchical function, there are two H-symbols, one represented by $\{0\}$ and second one being a pair of possible points $\{\pm 2\}$.

7.1.3 Hierarchical pairwise error probability

H-PEP for isomorphic NCM

H-PEP definition

Assume the H-message decoding metric is $\mu_b(x)$ and that the decoder *decision* processing *maximises* its value, $\hat{b} = \arg\max_b \mu_b(x)$. H-PEP is

$$P_{2H}(b'|b) = \Pr\left\{\mu_b(x) < \mu_{b'}(x)|b = \chi(\tilde{b}_x)\right\} \tag{7.1}$$

where $\tilde{b}_x$ are *actually* transmitted component data and $b \neq b'$ are some given H-messages. We will also use a simplified notation $P_{2H} = \Pr\{\mu_b < \mu_{b'}|b\}$.

The H-PEP is thus the probability that the H-metric for correct H-message b is smaller than the one for some other message b' provided that the received signal is consistent with b, i.e. all component transmitted data are such that $b = \chi(\tilde{b}_x)$. The form of the metric used in H-PEP is arbitrary. It does not even need to be the metric leading to the optimal performance. In such a case it would simply analyse the performance under that suboptimal metric and potentially suggest how to optimise the code for that given (suboptimal) metric. The most common example is the MAP metric.

H-PEP for isomorphic NCM

The evaluation of H-PEP related directly to the *message* level HNC map metric is a difficult task. It becomes much easier when the metric is related to the *code* level HNC map. It directly employs the codewords into the calculation which will provide better insight on what the code should optimally look like. If the NCM is *isomorphic* then $P_{2H}(b'|b) = P_{2H}(c'|c)$ where $c = \mathcal{C}(b)$ and $c' = \mathcal{C}(b')$. From now on, we will assume isomorphic NCM, and thus

$$P_{2H}(c'|c) = \Pr\{\mu_c(x) < \mu_{c'}(x)|c\}. \tag{7.2}$$

We will also assume the use of *MAP decoding metric.* The H-metric is a marginalisation

$$\mu_c(x) = p(x|c) = \frac{1}{p(c)} \sum_{\tilde{c}:c} p(x|\tilde{c})p(\tilde{c}) \tag{7.3}$$

where we used notation $(\tilde{c} : c) = \{\tilde{c} : \; c = \chi_c(\tilde{c})\}$. Notice a very important fact that the step from message H-PEP evaluation to codeword based one (which is trivial in a single user case) requires a *specific assumption and further treatment.*

In (7.2), we first focus on the conditioning by the received signal consistent with c. There are multiple of component codewords $\tilde{c}_x$ in the received signal consistent with

c $P_{2H} = \Pr\{\mu_c(x) < \mu_{c'}(x) | \bigcup_{\tilde{c}_x:c} \tilde{c}_x\}$. Events $\tilde{c}_x$ are *disjoint*. If we assume that they are also *equally probable* $\Pr\{\tilde{c}_x : c\} = \text{const}$ then[3]

$$P_{2H} = \frac{1}{M_{\tilde{c}:c}} \sum_{\tilde{c}_x:c} \Pr\left\{\mu_c(x) < \mu_{c'}(x) | \tilde{c}_x\right\} \tag{7.4}$$

where $M_{\tilde{c}:c}$ is the size of sub-codebook $\tilde{C}(c)$. $\tilde{C}(c)$ is a subset of $\tilde{C}$ where we take only the entries consistent with c. The H-PEP is the average over all consistent source node component codes. This is additional level of averaging over those being present in the traditional pairwise error probability calculation.

The most probable event

We can upper-bound the H-PEP by *the most probable pairwise H-constellation event*

$$P_{2H} \leq P_{2H_m} = \max_{\tilde{c}_x:c} \Pr\left\{\mu_c(x) < \mu_{c'}(x) | \tilde{c}_x\right\}. \tag{7.5}$$

Notice that the hierarchical codewords c, c' are still fixed. The overall error rate behaviour can be then (similarly as in classical single user code case) upper-bounded by the *overall* most probable pairwise event

$$P_{2H_{\max}} = \max_{c \neq c', \tilde{c}_x:c} \Pr\left\{\mu_c(x) < \mu_{c'}(x) | \tilde{c}_x\right\}. \tag{7.6}$$

H-PEP for Gaussian memoryless channel

Gaussian channel

In the next step, we will constrain the treatment to a special case of *Gaussian memoryless channel.* The observation likelihood for *all component* codes is

$$p(x|\tilde{c}) = \frac{1}{\pi^m \sigma_w^{2m}} \exp\left(-\|x - u(\tilde{c})\|^2 / \sigma_w^2\right),$$

where m is a complete dimensionality of the signals (both per-symbol and length of the message), σ_w^2 is the variance per dimension, and $u(\tilde{c})$ is noiseless *constellation space point* observed at the receiver. Notice that channel model inherently contained in $u(\tilde{c})$ can be *arbitrary.*[4] *H-constellation* is the set of $u(\tilde{c})$ consistent with c, i.e. $\mathcal{U}(c) = \{u(\tilde{c}) : c = \chi_c(\tilde{c})\}$ where we properly count for multiplicities.

[3] For multiple *disjoint* and *equally probable* events B_i, it holds $\Pr\left\{A | \bigcup_{i=1}^{M} B_i\right\} = \frac{1}{M} \sum_{i=1}^{M} \Pr\{A|B_i\}$.

[4] In a special case of a linear flat fading channel, used now *only* as an example and not needed for the rest of the derivation, it would be $u = \sum_k h_k c_k$.

On top of assuming uniformly distributed $\tilde{c}$, $\Pr\{\tilde{c}\} = 1/M_{\tilde{c}}$, $M_{\tilde{c}} = |\tilde{\mathcal{C}}|$, we also assume ***uniform*** c, i.e. $\Pr\{c\} = \text{const} = 1/M_c$ where $M_c = |\mathcal{C}|$ is the size of H-codebook. Then the metric is

$$\mu_c = \frac{M_c}{M_{\tilde{c}}} \sum_{\tilde{c}:c} p(x|\tilde{c}) = \frac{M_c}{M_{\tilde{c}}} \frac{1}{\pi^m \sigma_w^{2m}} \sum_{\tilde{c}:c} e^{-\frac{1}{\sigma_w^2}\|x-u(\tilde{c})\|^2} \tag{7.7}$$

and its normalised form $\acute{\mu}_c = \mu_c \pi^m \sigma_w^{2m} M_{\tilde{c}}/M_c$ is

$$\acute{\mu}_c = \sum_{\tilde{c}:c} \exp\left(-\|x - u(\tilde{c})\|^2/\sigma_w^2\right). \tag{7.8}$$

H-distance

The H-distance for the complete message is $u^{\text{Hmin}}(c) = \arg\min_{u(\tilde{c}):\chi_c(\tilde{c})=c} \|x - u(\tilde{c})\|^2$. We can factorise (7.8)

$$\acute{\mu}_c = e^{-\frac{1}{\sigma_w^2}\|x-u^{\text{Hmin}}(c)\|^2} \sum_{\tilde{c}:c} e^{-\frac{1}{\sigma_w^2}\left(\|x-u(\tilde{c})\|^2 - \|x-u^{\text{Hmin}}(c)\|^2\right)} \tag{7.9}$$

where all differences in the summation are *non-negative* $\|x-u(\tilde{c})\|^2 - \|x-u^{\text{Hmin}}(c)\|^2 \geq 0$. Finally, taking the negative scaled logarithm $\rho_c = -\sigma_w^2 \ln \acute{\mu}_c$, we get the decoder metric

$$\rho_c = \|x - u^{\text{Hmin}}(c)\|^2 - \sigma_w^2 \eta_c \tag{7.10}$$

where the correction term is

$$\eta_c = \ln \sum_{\tilde{c}:c} e^{-\frac{1}{\sigma_w^2}\left(\|x-u(\tilde{c})\|^2 - \|x-u^{\text{Hmin}}(c)\|^2\right)}. \tag{7.11}$$

Clearly, it holds $0 \leq \eta_c \leq \ln(M_{\tilde{c}}/M_c)$. The correction term is zero $\eta_c = 0$ if exactly one H-constellation point is the minimal H-distance point $u^{(1)}(\tilde{c}) = u^{\text{Hmin}}(c)$ and all others (if any) are at much larger distance, $\forall i \neq 1$, $\|x - u^{(i)}(\tilde{c})\|^2 \gg \|x - u^{\text{Hmin}}(c)\|^2$. Notice that many practical component constellations, e.g. 2 component BPSK and XOR HNC with H-constellation $\{0, \{-2, 2\}\}$, do not fit under these conditions. On the other side, if for all $\tilde{c} : c$ the H-constellation point is the minimal H-distance point, then the correction is non-zero but *constant* and *independent* of x, $\eta_c = \ln(M_{\tilde{c}}/M_c)$. The non-zero value of η_c on its own does not present a problem from the H-PEP evaluation point of view. However its dependence on x is a problem. The presence and behaviour of the correction term also nicely demonstrates that the pure H-distance is *not* generally optimal decoding metric.

H-PEP

The H-PEP for a given set of messages and for c-consistent received signal ($\chi_c(\tilde{c}_x) = c$) is then

$$\begin{aligned} P_{2H_e} &= \Pr\{\rho_c(x) > \rho_{c'}(x)|\tilde{c}_x\} \\ &= \Pr\{\|x - u^{\mathrm{Hmin}}(c)\|^2 - \|x - u^{\mathrm{Hmin}}(c')\|^2 - \sigma_w^2 \eta_{c,c'} > 0|\tilde{c}_x\} \end{aligned} \tag{7.12}$$

where $\eta_{c,c'} = \eta_c - \eta_{c'}$. The inequality is reversed since we used negative scaled logarithm metric. Let us now denote the c-consistent noiseless part of the received signal for given $\tilde{c}_x$ as $u(\tilde{c}_x)$. It has to be a member of the H-constellation set for the c H-symbol, i.e. $u(\tilde{c}_x) \in \mathcal{U}(c)$. The condition of c-consistent received signal is thus reflected in having $x = u(\tilde{c}_x) + w$ and consequently

$$\begin{aligned} P_{2H_e} = \Pr\Big\{&\|u(\tilde{c}_x) + w - u^{\mathrm{Hmin}}(c)\|^2 \\ &- \|u(\tilde{c}_x) + w - u^{\mathrm{Hmin}}(c')\|^2 - \sigma_w^2 \eta_{x,c,c'} > 0\Big\} \end{aligned} \tag{7.13}$$

where the correction terms under this condition are $\eta_{x,c,c'} = \eta_{x,c} - \eta_{x,c'}$

$$\eta_{x,c} = \ln \sum_{\tilde{c}:c} e^{-\frac{1}{\sigma_w^2}\left(\|u(\tilde{c}_x)+w-u(\tilde{c})\|^2 - \|u(\tilde{c}_x)+w-u^{\mathrm{Hmin}}(c)\|^2\right)}, \tag{7.14}$$

$$\eta_{x,c'} = \ln \sum_{\tilde{c}':c'} e^{-\frac{1}{\sigma_w^2}\left(\|u(\tilde{c}_x)+w-u(\tilde{c}')\|^2 - \|u(\tilde{c}_x)+w-u^{\mathrm{Hmin}}(c')\|^2\right)}. \tag{7.15}$$

The expression of the distances difference that appears in P_{2H_e} (and with minor modification in $\eta_{x,c}, \eta_{x,c'}$) can be further manipulated ($\langle \centerdot ; \centerdot \rangle$ denotes inner product)

$$\begin{aligned} &\|u(\tilde{c}_x) + w - u^{\mathrm{Hmin}}(c)\|^2 - \|u(\tilde{c}_x) + w - u^{\mathrm{Hmin}}(c')\|^2 \\ =&\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2 - \|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2 \\ &+ 2\mathrm{Re}\left[\left\langle u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c); w\right\rangle\right] - 2\mathrm{Re}\left[\left\langle u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c'); w\right\rangle\right] \\ =&\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2 - \|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2 \\ &- 2\mathrm{Re}\left[\left\langle u^{\mathrm{Hmin}}(c) - u^{\mathrm{Hmin}}(c'); w\right\rangle\right]. \end{aligned} \tag{7.16}$$

$\xi = -2\mathrm{Re}\left[\left\langle u^{\mathrm{Hmin}}(c) - u^{\mathrm{Hmin}}(c'); w\right\rangle\right]$ is Gaussian real-valued scalar zero-mean random variable with the variance $\sigma_\xi^2 = 2\sigma_w^2 \|u^{\mathrm{Hmin}}(c) - u^{\mathrm{Hmin}}(c')\|^2$. Then

$$P_{2H_e} = \Pr\left\{\xi > \|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2 - \|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2 + \sigma_w^2 \eta_{c,c'}\right\}. \tag{7.17}$$

A similar manipulation can be done for the correction terms

$$\eta_{x,c} = \ln \sum_{\tilde{c}:c} e^{-\frac{1}{\sigma_w^2}\left(\|u(\tilde{c}_x)-u(\tilde{c})\|^2 - \|u(\tilde{c}_x)-u^{\mathrm{Hmin}}(c)\|^2 - 2\mathrm{Re}\left[\left\langle u(\tilde{c})-u^{\mathrm{Hmin}}(c); w\right\rangle\right]\right)}, \tag{7.18}$$

$$\eta_{x,c'} = \ln \sum_{\tilde{c}':c'} e^{-\frac{1}{\sigma_w^2}\left(\|u(\tilde{c}_x)-u(\tilde{c}')\|^2 - \|u(\tilde{c}_x)-u^{\mathrm{Hmin}}(c')\|^2 - 2\mathrm{Re}\left[\left\langle u(\tilde{c}')-u^{\mathrm{Hmin}}(c'); w\right\rangle\right]\right)}. \tag{7.19}$$

Hierarchical distance and self-distance spectrum

The properties of the quantities determining the H-PEP clearly depend on two types of the H-constellation/codeword distances. The first one is the distance between the points belonging to *different* H-symbols and the second one is the distance between the points belonging to the *same* H-symbol. For this purpose, we define hierarchical distance and self-distance spectrum.

Hierarchical distance (H-distance) spectrum is a set

$$\mathcal{S}_H(c,c') = \left\{ \scriptstyle \|u(\tilde{c}) - u(\tilde{c}')\|^2 : c = \chi_c(\tilde{c}) \neq c' = \chi_c(\tilde{c}') \right\}. \tag{7.20}$$

We also define $\mathcal{S}_H = \bigcup_{c,c'} \mathcal{S}_H(c,c')$.

Hierarchical self-distance (H-self-distance) spectrum is a set

$$\mathcal{S}_{\bar{H}}(c) = \left\{ \scriptstyle \|u(\tilde{c}^{(a)}) - u(\tilde{c}^{(b)})\|^2 : c = \chi_c(\tilde{c}^{(a)}) = \chi_c(\tilde{c}^{(b)}) \wedge \tilde{c}^{(a)} \neq \tilde{c}^{(b)} \right\}. \tag{7.21}$$

We also define $\mathcal{S}_{\bar{H}} = \bigcup_c \mathcal{S}_{\bar{H}}(c)$.

NCM design rules based on H-PEP

We use (7.17,7.18,7.19) to establish qualitative design rules for NCM that minimise H-PEP. The situation is however less straightforward than in the classical single user code. There are several observations we need to keep in our mind before we start. There are multiple mutually correlated random variables in the expression and these cannot be easily factorised into a single one as in single-user code case. All $\xi, \eta_{x,c}, \eta_{x,c'}$ directly depend on Gaussian noise w and are continuous valued correlated variables. But also the hierarchical minimum distance points $u^{\mathrm{Hmin}}(c), u^{\mathrm{Hmin}}(c')$ depend on the received signal and therefore also on w. These variables are however discrete one. There are random and dependent on w but *constrained* to be inside the H-constellation and their influence on H-PEP can be thus controlled through the H-distance and H-self-distance spectrum. In order to *minimise* H-PEP, we should consider the following.

(1) The distance $\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2$ in (7.17) should be as large as possible. Notice that $\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2 \in \mathcal{S}_H(c,c')$ for arbitrary noise w realisation.

(2) The self-distance $\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2$ in (7.17) should be as small as possible. Notice that $\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2 \in \mathcal{S}_{\bar{H}}(c)$ for arbitrary noise w realisation.

(3) The variance of the ξ variable is proportional to $\|u^{\mathrm{Hmin}}(c) - u^{\mathrm{Hmin}}(c')\|^2 \in \mathcal{S}_H(c,c')$ which is constrained by the H-distance spectrum.

(4) The correction term $\eta_{x,c,c'}$ should be as large as possible which in turn means maximising $\eta_{x,c}$ and minimising $\eta_{x,c'}$. Behaviour of $\eta_{x,c}$ is dictated by H-self-distance spectrum while the behaviour of $\eta_{x,c'}$ is jointly dictated by both H-distance and H-self-distance spectrum.

(5) The maximum value of $\eta_{x,c}$ is $\ln(M_{\tilde{c}}/M_c)$ and it is reached when arguments of the exponentials are zero, i.e. when all self-distances are zero $\mathcal{S}_{\bar{H}}(c) = \{0\}$. All H-constellation (codeword) points for given c are *identical*. We will call this a *self-folded* H-constellation (codebook) or *self-folded NCM*. If the H-constellation/codebook is self-folded then the arguments of the exponentials in $\eta_{x,c'}$ are also all zeros, and thus $\eta_{x,c'} = \ln(M_{\tilde{c}}/M_c)$ and the overall correction term is zero regardless of the noise $\eta_{x,c,c'} = 0$. Self-folded NCM also causes the self-distance in (7.17) to be zero, and thus

$$P_{2H_e}^{\mathrm{SF}} = \Pr\left\{\xi > \|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2\right\}. \tag{7.22}$$

(6) Now let us have a look at the situation when the NCM is *not* self-folded. Let us assume that the spread in self-distances is symmetric for all c. If it was not a symmetric one then the case that would make an advantage for $P_{2H_e}(c'|c)$ would become a disadvantage for $P_{2H_e}(c|c')$ in terms of the possible compensation as discussed in point (6b) below.

(6a) Let us also assume that some point pair $\tilde{c}_x, \tilde{c}$ in the H-constellation maximises the self-distance $\|u(\tilde{c}_x) - u(\tilde{c})\|^2$ to some particular value $d_{\bar{H}}^2$. The expression $\eta_{x,c}$ will not be the maximal one (as for self-folded case) but it will be somewhat smaller. For the given pair of points, the argument $\|u(\tilde{c}_x) - u(\tilde{c})\|^2$ of the exponential in (7.18) increases to the value $d_{\bar{H}}^2$. The second term $\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2$ will highly likely (at least for high SNR) be zero since the minimum H-distance point is the closest to the received signal. The degradation of the first noiseless term in (7.18) is thus $d_{\bar{H}}^2$ at least for that given point pair.

(6b) This degradation can be possibly compensated by the improvement in the term (7.19). In the most favorable case for the improvement, the points $u(\tilde{c}_x), u(\tilde{c}'), u^{\mathrm{Hmin}}(c')$ lie in the line and the maximal value of $\|u(\tilde{c}_x) - u(\tilde{c}')\|^2 - \|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2$ is $d_{\bar{H}}^2$. Where, by the assumption of the symmetry, points $u(\tilde{c}'), u^{\mathrm{Hmin}}(c')$ are constrained to the distance $\|u(\tilde{c}') - u^{\mathrm{Hmin}}(c')\|^2 = d_{\bar{H}}^2$. So the noiseless terms in the exponentials of (7.19) can, at the best, just compensate the degradation of the argument of (7.18) but practically it will be even worse.

(6c) The noise terms in both (7.18) and (7.19), i.e.

$$2\mathrm{Re}\,\left[\left\langle u(\tilde{c}) - u^{\mathrm{Hmin}}(c); w\right\rangle\right]$$

and

$$2\mathrm{Re}\,\left[\left\langle u(\tilde{c}') - u^{\mathrm{Hmin}}(c'); w\right\rangle\right]$$

are given by the self-distances only. The left-hand sides in the inner products are different but under the assumption of symmetric self-distances $\mathcal{S}_{\bar{H}}(c) \approx \mathcal{S}_{\bar{H}}(c')$ they will make the noise term highly correlated, and thus both will be affecting the arguments of the exponentials in (7.18) and (7.19) the same way.

(6d) The main expression (7.17) also contains the self-distance. A positive value of $\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c)\|^2$ decreases the right-hand side of the inequality and increases the H-PEP.

(6e) As we see, the nonzero spread of the self-distances *cannot* improve the H-PEP and will make highly likely the things only worse.

7.1.4 Summary

The analysis of the H-PEP behaviour lead us to the final main result identifying the self-folded NCM design criterion.

Conjecture: Self-folded NCM (H-constellation/codebook) minimises H-PEP

Assume isomorphic NCM in Gaussian memoryless channel, decoding MAP H-metric, and uniform component messages and HNC map such that $\Pr\{\tilde{c}\} = 1/M_{\tilde{c}}$, $\Pr\{c\} = 1/M_c$, $\Pr\{\tilde{c} : c\} = M_c/M_{\tilde{c}}$. Self-folded NCM, i.e. the one with zero H-self-distance spectrum $\mathcal{S}_{\bar{H}} = \{0\}$, minimises H-PEP which is (Q is complementary Gaussian CDF)

$$P_{2H_e}^{\mathrm{SF}} = Q\left(\sqrt{\|u(\tilde{c}_x) - u^{\mathrm{Hmin}}(c')\|^2/(2\sigma_w^2)}\right). \tag{7.23}$$

It is important to note that the self-folding property is expected to be *natural*, i.e. naturally performed by the channel combining the component signals into the H-constellation also fully respecting the channel parameterisation. Notice that the modulo-lattice preprocessing (used in CaF) achieves the hierarchical self-folding but it achieves that by the *force*. The price paid for this enforcement is the distortion of the noise which becomes modulo-equivalent Gaussian.

8 Practical encoding and decoding of CaF based NCM

8.1 Complex low density lattice codes to physical layer network coding

CaF, lattice network coding (LNC) and the aforementioned multilevel lattice network coding (MLNC) lay the theoretical foundations for achieving high-throughput WPLNC-relaying based on any lattice codes, and MLNC also gives the practical design guideline. Lattice codes can be constructed by the existing non-binary channel codes, e.g. Construction A or D lattices, and these lattice codes have been designed in LNC. In this section, we consider an alternative design to LNC, where the recently developed complex low density lattice codes (CLDLC) are employed. Differing from construction A or D, CLDLC is directly designed in the Euclidean space, and has good algebraic properties, which are well suitable for LNC. Especially CLDLC is in principle capable of improving the overall LNC performance and throughput compared to the traditional constructions, due to its high coding gain.

8.1.1 Introduction

WPLNC was introduced around 2006 independently by several research groups [47] [121] [22]. It has been shown to be very effective in improving the throughput of a two way relay channel (2-WRC). The core idea is that the intermediate relay attempts to infer and forward linear combinations of the simultaneously received signals, instead of decoding the transmitted signals individually. Nazer and Gastpar [22] proposed a new approach to WPLNC, namely compute-and-forward (CaF) which extends the 2-WRC to a more general network topology. In more recent work, Feng *et al.* [49] formulated a more general algebraic framework for lattice-based WPLNC, namely lattice network coding (LNC), which relates CaF to the fundamental theorem of finitely generated modules over a principal ideal domain (PID). One possible solution for LNC design is to employ existing linear channel codes to construct lattices, e.g. [61] [122]. Another solution for LNC design is to employ practical high coding gain lattice codes where the lattice codes are directly designed in the Euclidean space. Sommer *et al.* developed practically decodable lattice codes, e.g. signal codes [74], which has been applied in WPLNC [123].

An alternative is provided however by the use of Low density lattice codes (LDLC) [18] in LNC. LDLC has high coding gain in comparison to signal codes, and manageable decoding complexity based on parametric belief propagation decoding. Due to its algebraic properties and high coding gain, LDLC shows good performance in CaF relaying [124].

To improve the performance further, we construct low density R-lattices where R denotes a principal ideal domain, and hence the generated lattice forms a special R-module which is closely associated with the complex low density lattice generator which further defines the complex low density lattice codes (CLDLC) [73]. Note that CLDLC is not simply a 2-dimensional real LDLC, but rather the low density R-lattices are directly generated over the domain of complex numbers. The performance of CLDLC outperforms 2-dimensional real LDLC since real parity check matrix normally suffers from short loops. Compared to real LDLC, CLDLC is not merely a more practically feasible code which adapts to channel fading and gives in principle better performance at low dimensions; more importantly, the lattices constructed via CLDLC form precisely a finitely generated R-module over PID. This algebraic property makes it especially suitable for LNC design, and effectively improves the overall rate compared to the real LDLC-based CaF. We extend the parametric BP decoding algorithm [125] [126] to the complex case, and propose a Gaussian mixture reduction model over the complex domain. We present the low density R-lattice-based LNC design based on some algebraic theorems.

Note that CLDLC is itself a lattice code directly designed in the Euclidean space, which improves the network throughput at manageable complexity, and is more convenient when compared to the lattices constructed from linear codes, for which, for example a large prime field may be required. Our work here shows the remarkable potential of CLDLC used in LNC, and may motivate further research in this area.

Notation definitions

We use $\mathbb{C}$ and $\mathbb{Z}$ to denote the fields of complex numbers and integers, respectively. $\mathbb{F}_q$, $q > 1$, $q \in \mathbb{Z}$ denotes the finite field with size q. We also use boldface lowercase and boldface uppercase to denote column vector and matrices, respectively, e.g. $\mathbf{h} = [h_1, \cdots, h_n]^T$. $\mathbf{h}_i$ and $\mathbf{h}^i$ denotes the i^{th} row and the i^{th} column of the matrix $\mathbf{H}$. We also denote $\boldsymbol{h}_{\setminus j} = [h_1, \cdots, h_{j-1}, h_{j+1}, h_n]$. $\mathbb{F}^n_{\mathbf{q}}$ denotes the direct product of n finite fields, where the size of the j^{th} field $j \in \{1, 2, \cdots, n\}$ is determined by $q_j \in \mathbb{Z}$.

8.1.2 Complex low density lattice codes

We denote R as a principal ideal domain (PID), then a Low Density R-Lattice is defined as:

Definition 1 (Low density R-lattices): An n-dimensional low density R-lattice Λ is constructed as a set of R-linear combinations of n linearly independent column vectors in a low density lattice generator matrix $\mathbf{G}_\Lambda \in \mathbb{C}^{N \times n}$ ($n \leq N$):

$$\Lambda = \{\mathbf{G}_\Lambda \mathbf{b} \mid \mathbf{b} \in R^n\}, \tag{8.1}$$

where $\mathbf{G}_\Lambda$ is non-singular and its inverse is a sparse matrix $\mathbf{H} = \mathbf{G}_\Lambda^{-1}$ (which is called the parity check matrix). Examples of PID include Gaussian integers $\mathbb{Z}[i]$ ($i = \sqrt{-1}$) and

Eisenstein integers $\mathbb{Z}[\omega]$ ($\omega = e^{i\frac{2\pi}{3}}$), which correspond to the low density $\mathbb{Z}[i]$-lattices and $\mathbb{Z}[\omega]$-lattices, respectively, following definition 1.

Unless otherwise stated, we assume $\mathbf{H}$ is a Latin square matrix in this paper, and hence $N = n$, such that each row and column have the same degree d. An (n, d) Latin square CLDLC can be constructed by designing a sparse parity check matrix $\mathbf{H}$ having constant row and column weight d. Let

$$\mathbf{h} = [1, \tau, \tau, \cdots, \tau, 0, \cdots, 0] \tag{8.2}$$

be a generating sequence with $(d-1)$ τs ($\tau \in \mathbb{C}$). $\mathbf{H}$ should be designed such that each row and column is a permutation of $\mathbf{h}$ followed by a random sign change, and of course $\mathbf{H}$ has to be cycle free. τ should be carefully designed to ensure fast convergence in the iterative BP decoder. Thus, $\alpha \triangleq \frac{\sum_{l=2}^{d}|h_l|^2}{|h_1|^2} \in (0, 1)$. We consider here $|\tau|^2 = \frac{1}{d}$. The volume of the Voronoi region $\mathcal{V}(\lambda)$ (Section 8.1.3) of CLDLC equals $\det(\mathbf{G}_\Lambda^\dagger \mathbf{G}_\Lambda)$, and in the sequel we normalise $\mathcal{V}(\lambda) = 1$.

The n-dimensional low-density R-lattice λ is transmitted through the complex additive white Gaussian noise channel $\mathbf{y} = \boldsymbol{\lambda} + \mathbf{n}$, where $\mathbf{n} \sim \mathcal{CN}(0, 2\sigma^2 \mathbf{I}^{n\times 1})$.

CLDLC decoder

The computational costs of the traditional sequential maximum likelihood detection $\widehat{\mathbf{b}} = \arg\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{G}\mathbf{b}\|^2$ is unaffordable. Hence, the iterative detection employing the belief propagation (BP) algorithm over the bipartite graph of $\mathbf{H}$ is a good trade-off between performance and complexities; the messages calculated in VNs and CNs are iteratively exchanged to achieve full convergence, and hence improve the reliability. Note that the metrics exchanged between the variable nodes (VN) and check nodes (CN) are continuous functions over $(-\infty, \infty)$ rather than scalar values, e.g. LLR in LDPC, which are closely related to the complex Gaussian distribution, expressed by a 2-dimensional Gaussian function:

$$\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{V}) = \frac{1}{2\pi\sqrt{|\mathbf{V}|)}} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m})^T \mathbf{V}^{-1}(\mathbf{z}-\mathbf{m})} \tag{8.3}$$

where $\mathbf{m}$ and $\mathbf{V}$ denote the real mean vector and covariance matrix. Large computational complexity and storage is needed to quantise the continuous function, and perform Fourier transforms for the convolution operations. The Gaussian mixture model is a good method to make the LDLC decoder practically feasible. Operations at CN and VN including convolution and multiplication are performed in the form of Gaussian mixture distributions and the output is another Gaussian mixture. A mixture of N complex Gaussians is represented by:

$$\mathcal{GM}(\mathbf{z}) = \sum_{i=1}^{N} a_i \mathcal{N}(\mathbf{z}; \mathbf{m}_i, \mathbf{V}_i) \tag{8.4}$$

where $a_i \geq 0$ satisfying $\sum_i^N a_i = 1$. It can be represented in parametric form as a list $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_1, \cdots, \mathcal{L}_N\}$, $\mathcal{L}_i = (\mathbf{m}_i, \mathbf{V}_i, a_i)$. At the check node, the i^{th} CN calculates function $\rho_j(\mathbf{z})$ (which is a Gaussian mixture) for the j^{th} VN connected to it, based on the $d-1$ functions $\boldsymbol{\mu}_{\backslash j}(\mathbf{z})$ transmitted from all VNs that are connected to this CN. In terms of the parity check equation, message $\rho_j(\mathbf{z})$ involves convolution operation between all elements in $\boldsymbol{\mu}_{\backslash j}(\mathbf{z})$ after stretching/expanding and rotating in terms of the complex components $\mathbf{h}_i$. The parity check outputs have to be an integer $\mathbb{Z}[\eta]$ which is unknown at the decoder, hence the periodic expansion with period $\frac{1}{|h_{i,j}|}$ is necessary. Inspired by the work of [126], we consider here that the messages output from CN and VNs are only single complex Gaussian function. Then the periodic extension step occurs at VNs rather than CNs.

Variable nodes. Assume the i^{th} VN computes $\mu_j(\mathbf{z})$ for the j^{th} CN connected to it, based on the input messages $\hat{\boldsymbol{\rho}}_{\backslash j}(\mathbf{z})$ which are single Gaussians. A small "trick" is that the periodic extension step creates a new Gaussian mixture such that each component is uniformly distributed. Suppose Gaussian integers are considered here, we have:

$$\rho_\ell(\mathbf{z}) = \frac{1}{k} \sum_{b \in \mathbb{Z}^k[i]} \mathcal{N}(\mathbf{z}; \mathbf{m}_\ell + \frac{b}{h_\ell}, \mathbf{V}_\ell) \tag{8.5}$$

where $\mathbb{Z}^k[i]$ denotes a set including k finite integers in $\mathbb{Z}[i]$. $\mathbb{Z}^k[i]$ should be restricted to those integers which are close to the channel messages. The variable output $\mu_j(\mathbf{z})$ is the product of all incoming messages, and also the channel message:

$$\mu_j(\mathbf{z}) = y(\mathbf{z}) \prod_{\ell=1, \ell \neq j}^{d} \rho_\ell(\mathbf{z}) \tag{8.6}$$

A forward-and-backward algorithm [126] can be used to reduce the computational complexity of message multiplication of $\mu_j(\mathbf{z})$. The forward and backward recursion is initialised with $\alpha_0(\mathbf{z}) = \gamma_0(\mathbf{z}) = \sqrt{y(\mathbf{z})}$, and the forward and backward recursion for the ℓ^{th} step is:

$$\overline{\alpha}_\ell(\mathbf{z}) = \mathcal{F}_{\text{M}}(\alpha_{\ell-1}(\mathbf{z}), \rho_\ell(\mathbf{z})) \tag{8.7}$$

$$\overline{\gamma}_\ell(\mathbf{z}) = \mathcal{B}_{\text{M}}(\alpha_{\ell+1}(\mathbf{z}), \rho_{\ell+1}(\mathbf{z})) \tag{8.8}$$

where $\mathcal{F}_{\text{M}}$ denotes the multiplication of the two Gaussian mixtures. The multiplication in parametric form $\mathcal{L}$ is detailed in [126]. Note that the number of mixed Gaussians in $\overline{\alpha}_\ell(\mathbf{z})$ grows exponentially as the iterations proceed. Hence, the Gaussian Mixture Reduction (GMR) [126] algorithm is developed to approximate a mixture of N Gaussians by another mixture of $N_{\max}$ Gaussians, $N_{\max} < N$. The GMR algorithm used for real LDLC compares the distance metric of all possible pairs of single Gaussians in the input list, and replaces the pair having the minimum metric by a single Gaussian using the second moment matching method. The single Gaussian with parameters $(\hat{\mathbf{m}}, \hat{\mathbf{V}})$ which gives the best approximation of L Gaussian mixtures should be selected according to:

$$(\hat{\mathbf{m}}, \hat{\mathbf{V}}) = \underset{\hat{\mathbf{m}},\hat{\mathbf{V}}}{\arg\min}\, \mathcal{K}(\mathcal{GM}(\mathbf{z})||\mathcal{N}(\mathbf{z}; \hat{\mathbf{m}}, \hat{\mathbf{V}})) \tag{8.9}$$

which minimises the Kullback-Leibler divergence $\mathcal{K}(\cdot||\cdot)$. This gives the estimate of parameters $(\hat{\mathbf{m}}, \hat{\mathbf{V}})$:

$$\hat{\mathbf{m}} = \sum_{\ell=1}^{L} a_\ell \mathbf{m}_\ell, \quad \hat{\mathbf{V}} = \sum_{\ell=1}^{L} a_\ell(\mathbf{V}_\ell + \mathbf{m}_\ell \mathbf{m}_\ell^T) - \hat{\mathbf{m}}\hat{\mathbf{m}}^T \tag{8.10}$$

Here we consider a GMR algorithm where an integer J is introduced, which determines the number of Gaussians combined into a 2-D single Gaussian. The details of the GMR algorithm are described in Algorithm 2. This algorithm reduces the number of loops for the greedy searches. In practice, we use the lower bound of the divergence, squared difference, instead of KLD. The number of Gaussians in $\overline{\alpha}_\ell(\mathbf{z})$ is limited to $N_{\max}$ by this algorithm, $\alpha_\ell(\mathbf{z}) = \mathrm{GMR}(\overline{\alpha}_\ell(\mathbf{z}))$. A large $N_{\max}$ gives better performance but the complexity increases. Here we found $N_{\max} = 3$ is a good trade-off between the complexity and performance for the parametric BP CLDLC decoder. The output $\mu_j(\mathbf{z})$ of the VN is a single Gaussian, given by:

$$\mu_j(\mathbf{z}) = \mathcal{M}(\mathrm{GMR}(\alpha_{j-1}(\mathbf{z}) \cdot \gamma_j(\mathbf{z})) \tag{8.11}$$

where $\mathcal{M}(\cdot)$ is the second moment matching defined in (8.9).

Check nodes. Since the periodic extension step is performed at the variable node rather than CN, this reduces the computational load at CN. Hence, at the i^{th} CN, $\rho_j(\mathbf{z})$ involves only convolution, stretching/expanding and rotation in terms of the complex coefficients $\mathbf{h}_i$, thus:

$$\hat{\rho}_j(\mathbf{z}) = \bigotimes_{\ell=1,\ell\neq j}^{d} \mu_\ell\left(-\frac{h_{i,\ell}}{h_{i,j}}\mathbf{z}\right) \tag{8.12}$$

where $\bigotimes$ denotes the convolution operation. Also the forward and backward algorithm is applied to simplify the calculation. If $\alpha(\mathbf{z})$ is a mixture with $N_{\max}$ Gaussians, the convolution $\mathcal{F}_{\mathrm{C}}\left(\alpha(\mathbf{z}), \mu_\ell(-\frac{h_{i,\ell}}{h_{i,j}}\mathbf{z})\right)$ produces another mixture of $N_{\max}$ Gaussians, with each component given by:

$$\mathbf{m}_p = \left[\mathrm{Re}\left(m_p^\alpha + \frac{h_{i,\ell}}{h_{i,j}} \cdot m_\ell^\mu\right), \mathrm{Im}\left(m_p^\alpha + \frac{h_{i,\ell}}{h_{i,j}} \cdot m_\ell^\mu\right)\right]^T \tag{8.13}$$

$$\mathbf{V}_p = \mathbf{V}_p^\alpha + \widetilde{\mathbf{H}}\mathbf{V}_\ell^\mu \widetilde{\mathbf{H}}, \qquad a_p = a_p^\alpha \tag{8.14}$$

where $\widetilde{\mathbf{H}} = \left[\mathrm{Re}\,(\frac{h_{i,\ell}}{h_{i,j}}), -\mathrm{Im}\,(\frac{h_{i,\ell}}{h_{i,j}}); \mathrm{Im}\,(\frac{h_{i,\ell}}{h_{i,j}}), \mathrm{Re}\,(\frac{h_{i,\ell}}{h_{i,j}})\right]$.

Algorithm 2 Modified GMR algorithm.

Input: A Gaussian Mixture with N components $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_N\}$, stopping parameters: θ and $N_{\max}$, and an appropriate integer value J

```
 1: do
 2:     $\mathcal{L}_{\mathrm{c}} \leftarrow \mathcal{L}$                                  ▷ $\mathcal{L}_{\mathrm{c}}$: current list stack
 3:     $\mathcal{L}_{\mathrm{c}} \leftarrow N$                                  ▷ $\mathcal{L}_{\mathrm{c}}$: length of $\mathcal{L}_{\mathrm{c}}$
 4:     $t \leftarrow 0$                                             ▷ $t$: counter
 5:     if $\binom{N-(J-1)t}{J} < \binom{N-t}{2}$ then
 6:         $k \leftarrow 2$
 7:     else
 8:         $k \leftarrow J$
 9:     end if
10:     $\theta_{\mathrm{c}} = \min_{\mathcal{L}_{\mathrm{c}}^k \in \mathcal{L}_{\mathrm{c}}} \mathcal{K}(\mathcal{L}_{\mathrm{c}}^k \parallel \mathcal{N}(\mathbf{z}; \hat{\mathbf{m}}, \hat{\mathbf{V}}))$      ▷ $\mathcal{L}_{\mathrm{c}}^k$: Arbitrary $k$ components in $\mathcal{L}_{\mathrm{c}}$.
11: while $\theta > \theta_{\mathrm{c}}$ or $N_{\mathrm{c}} > N_{\max}$
12: $t = t + 1$                                   ▷ : the $(t+1)^{\mathrm{th}}$ search loops next time
13: $\mathcal{L}_{\mathrm{c}}^k \leftarrow \arg\min_{\mathcal{L}_{\mathrm{c}}^k \in \mathcal{L}_{\mathrm{c}}} \mathcal{K}(\mathcal{L}_{\mathrm{c}}^k \parallel \mathcal{N}(\mathbf{z}; \hat{\mathbf{m}}, \hat{\mathbf{V}}))$      ▷ Find $k$ components which give the
    minimum divergence.
14: $\mathcal{L}_{\mathrm{c}} \leftarrow \left(\mathcal{L}_{\mathrm{c}} \bigcup \mathcal{M}(\mathcal{L}_{\mathrm{c}}^k)\right) \setminus \left\{\mathcal{L}_{\mathrm{c}}^k\right\}$      ▷ $\mathcal{M}(\mathcal{L}_{\mathrm{c}}^k)$: Moment matching to a single Gaussian;
    $\setminus \left\{\mathcal{L}_{\mathrm{c}}^k\right\}$: delete $\left\{\mathcal{L}_{\mathrm{c}}^k\right\}$ from $\mathcal{L}_{\mathrm{c}}$.
15: $N_c \leftarrow N_c - (J-1)t$                              ▷ new length of $\mathcal{L}_{\mathrm{c}}$
16: Repeat 5–15.
17: Output list $\mathcal{L}_{\mathrm{c}}$ is a mixture of $N_{\max}$ Gaussians.
```

8.1.3 Practical WPLNC via CLDLC

Definitions and algebraic framework

Definition 2 (Quantiser): A lattice quantiser, $\mathcal{Q}_\Lambda$: $\mathbb{C}^n \rightarrow \Lambda$, maps a point $\mathbf{s}$ in $\mathbb{C}^n$ to the nearest point in Λ in Euclidean distance:

$$\mathcal{Q}_\Lambda(\mathbf{s}) = \arg\min_{\lambda \in \Lambda} \|\mathbf{s} - \lambda\| \tag{8.15}$$

Definition 3 (Voronoi region): The Voronoi region of a lattice point λ, denoted by $\mathcal{V}(\lambda)$, is the set of points in $\mathbb{R}^n$ closest to this point, i.e.,

$$\mathcal{V}(\lambda) = \{\mathbf{s} : \mathbf{s} \in \mathbb{R}^n, \mathcal{Q}_\Lambda(\mathbf{s}) = \lambda\} \tag{8.16}$$

The *fundamental Voronoi region* is defined as $\mathcal{V}(0)$, or simply represented as $\mathcal{V}$. Let $\mathrm{Vol}(\mathcal{V})$ denote the volume of $\mathcal{V}$ and $\mathrm{Vol}(\mathcal{V}) = \mathrm{Vol}(\Lambda)$.

Definition 4 (Goodness of lattices): Let $\mathbf{z}$ be an n-dimensional i.i.d. Gaussian vector, $\mathbf{z} \backsim \mathcal{N}(0, \sigma^2 \mathbf{I}^{n \times n})$, a sequence of lattices Λ is *Poltyrev-good* if

$$\Pr(\mathbf{z} \notin \mathcal{V}) \leq e^{-nE_p(\gamma)} \tag{8.17}$$

where $E_p(\,\cdot\,)$ is the Poltyrev exponent and γ is the volume-to-noise ratio (VNR),

$$\gamma = \frac{\mathrm{Vol}(\Lambda)^{\frac{2}{n}}}{2\pi e\sigma^2}. \tag{8.18}$$

Let Λ' denote the R-sublattice of Λ, then Λ/Λ' represents a quotient R-module. We refer to $\langle\varpi\rangle$ as the ideal generated by a single non-zero and non-unit component in R, and hence $R/\langle\varpi\rangle$ is a quotient ring. When ϖ is a prime in R, the coset representatives of $R/\langle\varpi\rangle$ form a field.

Let $\boldsymbol{\varpi}_{\mathbf{I}}$ denote a diagonal matrix:

$$\boldsymbol{\varpi}_{\mathbf{I}} = \begin{pmatrix} \varpi_{\mathbf{I},1} & 0 & 0 & 0 \\ 0 & \varpi_{\mathbf{I},2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \varpi_{\mathbf{I},n} \end{pmatrix} \tag{8.19}$$

where $\varpi_{\mathbf{I},i} \in R$, $i = 1, 2, \cdots, n$, and the low density R-sublattice Λ' is generated by:

$$\Lambda' = \underbrace{\boldsymbol{\varpi}_{\mathbf{I}}\mathbf{G}_{\Lambda}}_{\mathbf{G}_{\Lambda'}}\mathbf{b}, \quad \mathbf{b} \in R^n \tag{8.20}$$

where $\mathbf{G}_{\Lambda'}$ is referred to as the sublattice generator. The following theorem states the design criterion for CLDLC based WPLNC:

Theorem 8.1. *If $\forall i \in \{1, 2, \cdots, n\}$, the non-unit non-zero element $\varpi_{\mathbf{I},i}$ in R is a prime, and the cardinality $|R/\langle\varpi_{\mathbf{I},i}\rangle| = q_i$, then,*

$$\begin{aligned} \Lambda/\Lambda' &\cong R/\langle\varpi_{\mathbf{I},1}\rangle \times R/\langle\varpi_{\mathbf{I},2}\rangle \times \cdots \times R/\langle\varpi_{\mathbf{I},n}\rangle & (8.21)\\ &\cong \mathbb{F}_{q_1} \times \mathbb{F}_{q_2} \times \cdots \times \mathbb{F}_{q_n} = \mathbb{F}_{\mathbf{q}}^n & (8.22) \end{aligned}$$

Theorem 8.1 can be proved following two algebraic theorems. They are the *first isomorphism theorems for finitely generated modules* [63] where the submodule Λ' is the kernel of the linear mapping $\varphi : \Lambda \to R/\langle\varpi_{\mathbf{I},1}\rangle \times R/\langle\varpi_{\mathbf{I},2}\rangle \times \cdots \times R/\langle\varpi_{\mathbf{I},n}\rangle$; and the structure theorem of the finitely generated torsion R-module in elementary divisor form. □

Thus, Λ/Λ' is isomorphic to an n-dimensional finite field $\mathbb{F}_{\mathbf{q}}^n$, and hence is WPLNC-compatible [49]. Based on this, there exists a linear labeling $\varphi : \Lambda \to R/\langle\varpi_{\mathbf{I},1}\rangle \times R/\langle\varpi_{\mathbf{I},2}\rangle \times \cdots \times R/\langle\varpi_{\mathbf{I},n}\rangle$, which is a surjective R-module homomorphism whose kernel is Λ', and there also exists an injective map $\varphi^{-1} : (r_1 + \langle\varpi_{\mathbf{I},1}\rangle, \cdots, r_n + \langle\varpi_{\mathbf{I},n}\rangle) \to \Lambda$ satisfying $\varphi(\varphi^{-1}(\mathbf{w})) = \mathbf{w}$.

Encoding and decoding

Now we are able to describe the basic encoding and decoding operations for lattice code-based WPLNC. Based on Theorem 8.1, there exists an isomorphism between the quotient R-module Λ/Λ' and $\mathbb{F}_{\mathbf{q}}^n$. First we construct a surjective R-module homomorphism $\sigma : R^n \to \mathbb{F}_{\mathbf{q}}^n$, with its kernel equal to $((\varpi_{\mathbf{I},1}),(\varpi_{\mathbf{I},2}),\cdots,(\varpi_{\mathbf{I},n}))$, and also an injective map $\sigma^{-1} : \mathbb{F}_{\mathbf{q}}^n \to R^n$ satisfying $\sigma(\sigma^{-1}(\mathbf{w})) = \mathbf{w}$, $\forall \mathbf{w} \in \mathbb{F}_{\mathbf{q}}^n$. They show the explicit isomorphism between the message space and an R-module. Hence, we define $\varphi(\lambda) \triangleq \sigma(\mathbf{G}_\Lambda^{-1}\lambda)$ and the linear labelling $\varphi : \Lambda \to \mathbb{F}_{\mathbf{q}}^n$ which is also a surjective ring homomorphism. We further define $\varphi^{-1}(\mathbf{w}) = \mathbf{G}_\Lambda \sigma^{-1}(\mathbf{w})$ which is an injective map, and satisfies the bijection relationship $\varphi(\varphi^{-1}(\mathbf{w})) = \mathbf{w}$.

Encoding. The CLDLC codewords normally have large power as n increases. To limit the power expansion, the encoder should include a sublattice λ' where the generator matrix of Λ' is $\mathbf{G}_{\Lambda'} = \mathbf{G}_\Lambda \varpi_{\mathbf{I}}$. Following Theorem 8.1, the encoder $\mathcal{E}$ is designed by:

$$\mathbf{x} = \mathcal{E}(\mathbf{w}) = \varphi^{-1}(\mathbf{w}) - \mathbf{G}_{\Lambda'}\mathbf{k} \tag{8.23}$$

where $\mathbf{k} \in R^n$. The vector $\mathbf{k}$ is determined by the so-called shaping operations. In this paper we focus on the Tomlinson-Harashima shaping approach [74] which restricts the codewords $\mathbf{x}$ to a hypercube. Additional constraints need to be added into the parity check matrix $\mathbf{H}$ for CLDLC. Thus, $\mathbf{H}$ should have a lower-triangle form $\mathbf{H}_{\mathrm{L}}$ and the diagonal elements of $\mathbf{H}$ are all one. Note that in this case the top rows and right-hand columns have degree less than d. A more general approach to obtain the lower-triangular matrix is to use QR decomposition of the Latin square $\mathbf{H} = \mathbf{JQ}$ [127] where $\mathbf{J}$ is a lower triangular matrix and $\mathbf{Q}$ is orthonormal. Then each element of $\mathbf{k}$ can be obtained recursively by:

$$k_i = \left\lfloor \frac{\sigma^{-1}(w_i) - \sum_{l=1}^{i-1} J_{i,l}x_l}{\varpi_{\mathbf{I},i}} \right\rceil \tag{8.24}$$

$$x_i = \left(\sigma^{-1}(w_i) - \varpi_{\mathbf{I},i}k_i - \sum_{l=1}^{i-1} J_{i,l}x_l\right) / J_{i,i} \tag{8.25}$$

where θ denotes the Gaussian integer nearest to θ in the Euclidean distance. The encoder $\mathcal{E}$ establishes the relationship between the message space $\mathbb{F}_{\mathbf{q}}^n$ and low-density lattice partition Λ/Λ', and the power is conserved.

Decoding. The receiver aims to decode a linear combination $\mathbf{u} = \sum_{\ell=1}^{L} a_\ell \mathbf{w}_\ell$ over $\mathbb{F}_{\mathbf{q}}^n$ given the noisy, and faded low-density lattice codes $\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{n}$. We need to find the optimal scaling factor α and the corresponding coefficient vector $\mathbf{a}$ such that $\alpha\mathbf{y}$ is within the Voronoi region of a new low-density lattice point $\lambda = \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell$, and the expectation of the squared norm of the effective noise $\mathbf{n}_{\text{eff}} \triangleq \sum_{\ell=1}^{L}(\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}$ is

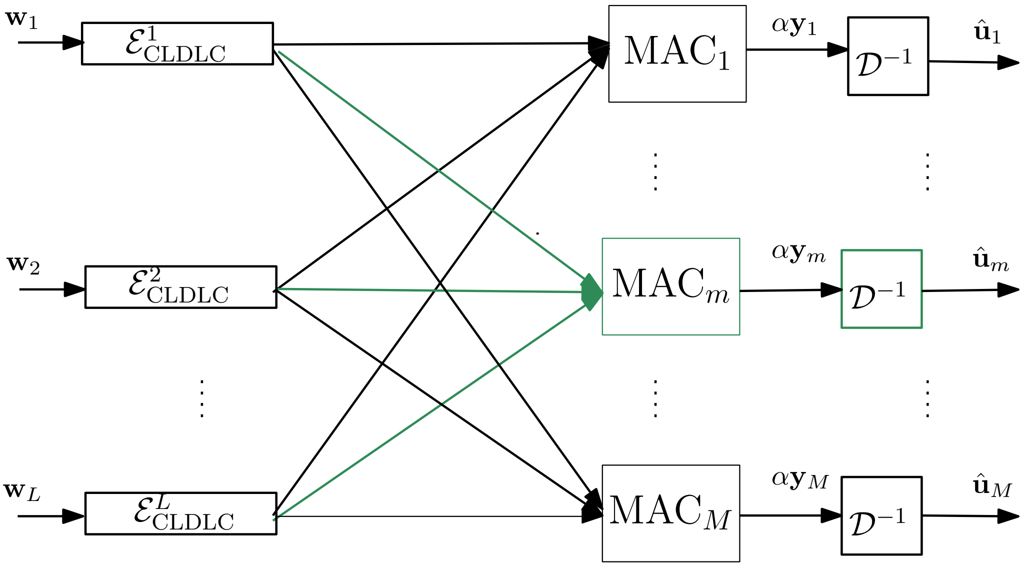


Figure 8.1: System diagram for CLDLC-WPLNC.

minimised. Finding the optimal values of α_{opt} and $\mathbf{a}_{\text{opt}}$ is equivalent to solving a shortest vector problem, as suggested in [49]. The estimates of $\hat{\mathbf{u}}$ are given by:

$$\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_\Lambda\left(\sum_{\ell=1}^{L} a_\ell\left(\varphi^{-1}(\mathbf{w}_\ell) - \mathbf{G}_{\Lambda'}\mathbf{k}_\ell\right) + \mathbf{n}_{\text{eff}}\right)\right) \tag{8.26}$$

$$\overset{(a)}{=} \varphi\left(\mathcal{Q}_\Lambda\left(\sum_{\ell=1}^{L} a_\ell\varphi^{-1}(\mathbf{w}_\ell) + \mathbf{n}_{\text{eff}}\right)\right) \tag{8.27}$$

$$\overset{(b)}{=} \varphi\left(\sum_{\ell=1}^{L} a_\ell\varphi^{-1}(\mathbf{w}_\ell)\right) + \varphi(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})) \tag{8.28}$$

$$\overset{(c)}{=} \sum_{\ell=1}^{L} \sigma(a_\ell)\mathbf{w}_\ell + \varphi(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})) \tag{8.29}$$

where (a) and (b) follows from the fact that $\mathbf{G}_{\Lambda'}\mathbf{k}_\ell$ is the kernel of φ, and the property of the ring homomorphism, respectively. From (8.29), $\mathbf{u}$ can be correctly recovered provided that $\varphi(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})) = 0$, thus, the effective noise should be within the Voronoi region of Λ'.

Computation rate

We now focus on the theoretical rate bound $\overline{\mathcal{R}}$ for the CLDLC-based relaying. Assume there are L sources, and we are concerned with the rate bound at the m^{th} relay. Hence, let the algebraic integer coefficient vector and channel coefficient vector be

$$\mathbf{a}_m = [a_{m,1}, a_{m,2}, \cdots, a_{m,L}]^T \in R^L$$

and

$$\mathbf{h} = [h_{m,1}, h_{m,2} \cdots, c_{m,L}] \in \mathbb{C}^L,$$

respectively. Then the scaled MAC output is:

$$\alpha_m \mathbf{y}_m = \sum_{\ell=1}^{L} a_{m,\ell}\mathbf{x}_\ell + \overbrace{\sum_{\ell=1}^{L}(\alpha h_{m,\ell} - a_{m,\ell})\mathbf{x}_\ell + \alpha_m \mathbf{z}_m}^{\text{effective noise}} \tag{8.30}$$

The source ℓ is subject to the power constraint $\boldsymbol{E}\left[\| \mathbf{x}_\ell \|^2\right] \leq nP$ due to the hypercube shaping. For convenience, we define the *signal-to-noise ratio* $\gamma = P/N_0$. The computation rate has the form of the logarithm of the ratio between the transmit power and the upper bound of the effective noise variance. When $\mathbf{h_m}$ and $\mathbf{a}_m$ are known, we employ the Nazer-Gastpar computation rate [22] [49] to approximate the achievable rate of the low density $\mathbb{Z}[i]$-lattice-based LNC, which is written as

$$\overline{\mathcal{R}} = \max_{\alpha\in\mathbb{C}} \log_2\left(\frac{\gamma}{|\alpha|^2 + \gamma \parallel \alpha\mathbf{h}_m - \mathbf{a}_m \parallel^2}\right) \tag{8.31}$$

This is reasonable when P is the average power of CLDLC lattices in the shaping region. However, the scalar optimisation is a bit different. Finding the optimal scalar α is a convex optimisation problem, which should correspond to the minimum value of the denominator in (8.31). Hence, equation (8.31) can be changed to:

$$\overline{\mathcal{R}} = \log_2\left(\mathbf{a}_m\left(\mathbf{I}_\mathrm{L} - \mathbf{h}_m^\dagger\mathbf{h}_m\frac{\gamma}{\gamma \parallel \mathbf{h}_m \parallel^2 + 1}\right)\mathbf{a}_m^\dagger\right)^{-1} \tag{8.32}$$

where † denotes the Hermitian operation, and $\mathbf{I}_\mathrm{L}$ is the $L \times L$ identity matrix. A positive-definite matrix $\mathbf{P}$ can be constructed:

$$\mathbf{P} = \gamma\left(\mathbf{I}_\mathrm{L} - \mathbf{h}_m^\dagger\mathbf{h}_m\frac{\gamma}{\gamma \parallel \mathbf{h}_m \parallel^2 + 1}\right) \tag{8.33}$$

such that it has a Cholesky decomposition $\mathbf{P} = \mathbf{T}\mathbf{T}^\dagger$. Note that to ensure the successful recovery of the original messages $\mathbf{w}_\ell$, $\ell = 1, 2, \cdots, L$, from the linear combinations over $\mathbb{F}_\mathbf{q}^n$, based on the quotient low density R-lattices, the coefficient vector $\mathbf{a}_m \in R^L$ should be optimised in terms of:

$$\mathbf{a}_{m,\mathrm{opt}} = \underset{\mathbf{a}_m \neq \bigcup_{i=1}^{n}\langle\varpi_{\mathbf{I},i}\rangle}{\arg\min} \parallel \mathbf{a}_m\mathbf{T} \parallel^2 \tag{8.34}$$

then the rate bound for CLDLC network coding is:

$$\begin{aligned}&\overline{\mathcal{R}}_{\mathrm{CLDLC}}\\ &= \log_2\left(\mathbf{a}_{m,\mathrm{opt}}\left(\mathbf{I}_\mathrm{L} - \mathbf{h}_m^\dagger\mathbf{h}_m\frac{\gamma}{\gamma \parallel \mathbf{h}_m \parallel^2 + 1}\right)\mathbf{a}_{m,\mathrm{opt}}^\dagger\right)^{-1}\end{aligned} \tag{8.35}$$

subject to (8.34). Finding optimal $\mathbf{a}_{m,\mathrm{opt}}$ is equivalent to a vector optimisation problem. Now equation (8.34) gives the rate bound for CLDLC based network coding for a given signal-to-noise ratio.

8.1.4 Decodability and simulation results

Decodability

As mentioned above, to perform hypercube shaping for CLDLC, the parity check matrix has to have a lower-triangular form, which means that hypercube-shaped CLDLC has asymmetric protection over each dimension. The constellation size at a certain dimension corresponding to a less protected check equation has to be reduced. This is different from signal codes where the generator matrix has a Toeplitz structure which is close to lower-triangular [74], and hence the cardinality of the original information at each dimension is not additionally limited. We are mainly concerned with the choice of $\boldsymbol{\varpi}_{\mathbf{I}}$ such that the information at each dimension can be fully recovered.

Since the degree of a given $\mathbf{H}_{\mathrm{L}}$ increases from 1 to d, $\varpi_{\mathbf{I},i}$ should be selected to ensure decodability for each level. Thus, the field size $q_i = |R/(\varpi_{\mathbf{I},i})|$ should gradually increase. The optimal selection of $\varpi_{\mathbf{I},i}$ for $\mathbf{H}_{\mathrm{L}}$ to minimise rate loss is still an open problem, but it is not our main concern in this paper. For the extended hypercube shaping, we define a nominal rate ϖ, and $\varpi_{\mathbf{I},i}$ should be selected such that $|\varpi_{\mathbf{I},i}| \leq \lfloor |J_{i,i}\varpi| \rfloor$. The practical average rate at each dimension is smaller than the nominal rate, but we expect to reduce the rate loss through the careful design of $\varpi_{\mathbf{I},i}$. Assume $\varpi = 4$, and the range of $\lfloor |J_{i,i}\varpi| \rfloor = \{5, 4, 3, 2\}$ (not uniformally distributed). If $\varpi_{\mathbf{I},i}$ is restricted in the form of $\mathbb{Z}[i] = \{a|\ a \in \mathbb{Z}\}$, we can only choose the Gausssian prime $\varpi_{\mathbf{I},i} = 3$ for those dimensions corresponding to $\lfloor |J_{i,i}\varpi| \rfloor \geq 3$. If $\varpi_{\mathbf{I},i}$ can be selected to be any Gaussian prime, the increased flexibility will be beneficial to the rate improvement. The codeword $|x_i|$ is upper bounded by $\frac{\sqrt{2}L}{2}$, where L fulfils $|\varpi_{\mathbf{I},i}| = \lfloor |J_{i,i}L| \rfloor$ for given $\varpi_{\mathbf{I},i}$.

Simulations

Figure 8.2 shows the simulated performance of the symbol error rate (SER) against volume-to-noise ratio (VNR) for an n-dimensional CLDLC over the complex AWGN channel, $n = 100$, $N_{\max} = 3$ and $d = 5$. The decoding algorithm is based on Section 8.1.2. We compare the performance of CLDLC to a $2n$-dimensional real LDLC, with $N_{\max} = 3$ and $d = 5$. We have SER of 10^{-4} at 2.5 dB from channel capacity for $(100, 5)$ CLDLC, and there is an improvement around 0.7 dB in SER performance when compared to $(200, 5)$ real LDLC. Both employ 10 iterations. This reveals that CLDLC has better performance than LDLC at relatively low dimension.

We show and analyse the performance of our CLDLC-based WPLNC scheme via a two way relay channel. We focus mainly on the multiple access channel (MAC) by which the overall system performance is dominated. Simulations were carried out for the low density $\mathbb{Z}[i]$-lattice code with dimension $n = 100$. We set the nominal rate $\varpi = 4$. The two approaches of choosing $\varpi_{\mathbf{I},i}$ described in Section 8.1.4 result in the average transmission rate around $\mathcal{R} = 2.80$ bits/dimension and $\mathcal{R} = 3.27$ bits/dimension, respectively. For the real LDLC case, we set the nominal constellation size $\varpi = 8$. Based on the hypercube shaping, we can obtain an average transmission rate of $\mathcal{R} = 2.80$ bits/dimension. In this paper, we show simulations for $h_1 = h_2 = 1$, but the general case will be investigated in a subsequent paper.

Figure8.3 illustrates the SER performance against signal-to-noise ratio (SNR) over a two way relay channel. SNR is defined as $\mathrm{SNR} = \frac{\boldsymbol{E}[\|\mathbf{x}_i\|^2]}{2\sigma^2}$. It is observed that CLDLC of rate 2.80 bits/dimension reaches an SER around 10^{-5} at 14 dB, approximately 5 dB from the hypercube shaping capacity (HSC) [127]; whereas we have SER of 10^{-5} at 15 dB for CLDLC of rate 3.27 bits/dimension which is around 4 dB from HSC. It is obvious that there is 1 dB improvement for CLDLC of rate 3.27 bits/dimension over CLDLC of rate 2.80 bits/dimension. This confirms our viewpoint in Section 8.1.4; thus, by carefully designing the sublattice (which is determined by $\varpi_{\mathbf{I},i}$), we are capable of reducing the shaping loss. In comparison to Figure8.2, we observe that there are additional 2 dB and 1 dB performance losses for CLDLC with $\mathcal{R} = 2.80$ and $\mathcal{R} = 3.27$, respectively, compared to the single-user complex AWGN channel. This is due to the combined effects of the MAC and shaping. We also compare the performance of CLDLC-based LNC with LDLC-based CaF in Figure8.3. When we employ both $(100, 5)$ CLDLC and $(100, 5)$ LDLC, the CLDLC based scheme outperforms LDLC by around 8.5 dB at rate $\mathcal{R} = 2.80$ bits/dimension which is approximately the difference between the two HSCs. This shows good potential of CLDLC over LDLC and is what we expect for CLDLC. The symbol error probability is defined as $\Pr(\hat{\mathbf{u}} \neq \mathbf{u}) = \Pr(\varphi(\mathcal{Q}_\Lambda(\mathbf{n}_{\text{eff}})) \neq 0)$. Thus, the linear function $\mathbf{u}$ can be fully decoded iff the quantised effective noise is a sublattice point $\Lambda' = \mathbf{G}_{\Lambda'}\mathbf{b}$. Our simulations show that the proposed CLDLC-based WPLNC scheme gives good coding gain and SER performance over MAC.

8.1.5 Conclusions

We have proposed the low density R-lattices and demonstrated that CLDLC, which derives from these is a class of good lattice codes suitable for WPLNC. We have given a modified GMR model for iterative BP decoding of CLDLC, analysed WPLNC-compatibility, and given the comparison with HSC and with LDLC, which showed that CLDLC with hypercube shaping works well in a two way relay channel. We also discussed the choice of low density sublattices to reduce the rate loss.

8.2 Convolutional lattice encoding and decoding

8.2.1 Introduction

Constructing lattices from Forward Error Correction (FEC) codes has been a rather active field of research in the past, and has led to Constructions A, B, C, D etc. [54]. In

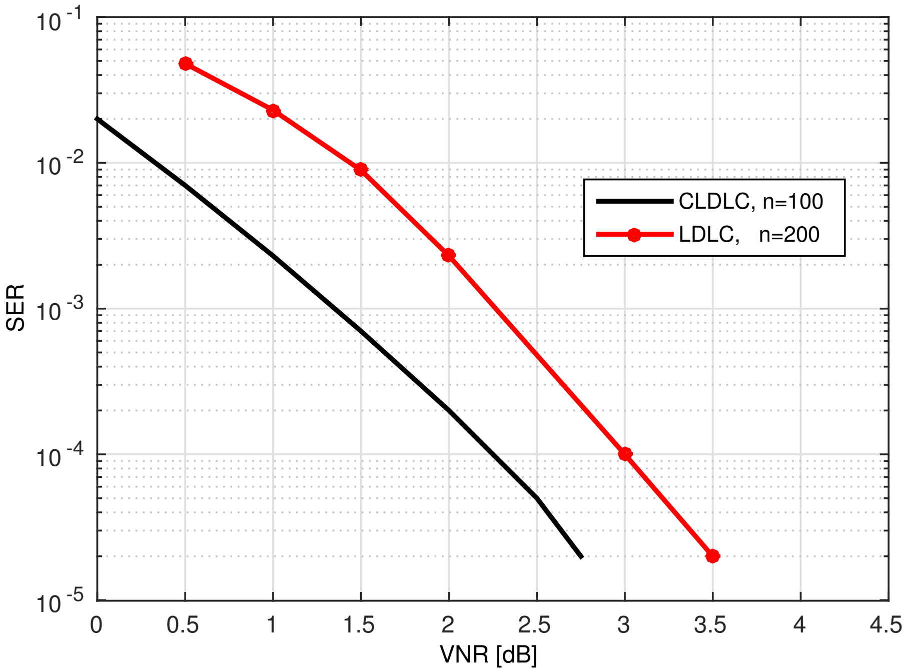


Figure 8.2: Symbol error rate for CLDLC based on the decoding algorithm in Section 8.1.2.

this section we are also interested in lattice construction, and in particular, Construction A because of its simplicity and Construction D because of the potential performance of the resulting codes. For lattice construction, we use convolutional codes as the underlying FEC code because capacity approaching Turbo codes consist of two (or more) convolutional codes; therefore, constructing convolutional lattices is a major step forward towards constructing Turbo lattices. There has been, surprisingly, little work on exploitation of convolutional codes for constructing lattices reported in the literature. Although [128, 129] discusses lattices based on convolutional codes (indeed, Turbo codes), the transmitted signals are restricted to be binary which loses the freedom to arbitrarily specify the rate of the lattice code: in this section we extend this to allow non-binary transmission with arbitrary transmission rate; moreover, lattice decoding algorithms have not been discussed in [128, 129], whereas we propose adopting the trellis structure of the underlying convolutional code for lattice decoding and demonstrate superior performance using this approach. This provides the possibility of implementing computationally feasible lattice decoding methods for convolutional lattices. Note that universal lattice decoding methods commonly applied in the literature for practical communication systems, e.g. sphere decoding, have until now been relatively complex, and as a result are applicable only to lattices with very short dimension [130–132]; for instance, a lattice decoder was proposed in [132] with relatively reasonable complexity that was examined for lattices of dimension up to 32. Indeed this is a major drawback be-

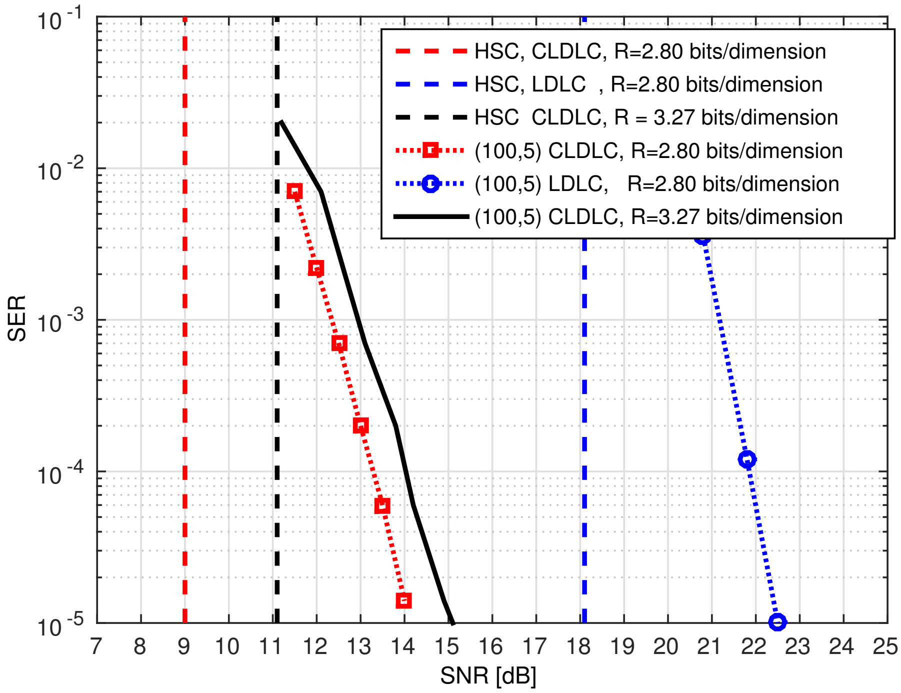


Figure 8.3: Symbol error rate for CLDLC-based WPLNC with fixed channel fading.

cause the code length (lattice dimension) of real communication systems is much longer than 32. Consequently, the lattice decoding method proposed in this section for decoding convolutional lattices is practically important due to its feasible complexity at high dimensions; furthermore, it will be observed in Sec 8.2.4 that the proposed lattice decoding approach significantly outperforms existing lattice decoding algorithms. [74] also studies convolutional lattices, however, the proposed scheme is mostly "attractive for Inter Symbol Interference (ISI) channels". Decoding algorithms of other ISI channel, in particular, Faster Than Nyquist (FTN) signalling has been studied by the authors in [133], however note that [74] considers code filters combined with ISI filters which results in unification of equalisation and decoding. In this section we are not interested in ISI channels nor FTN signalling but we would like to construct lattices from convolutional codes that are proved to approach capacity when applied in Turbo codes. Moreover, [74] considers single layer lattices whereas we assume multilayer as well as single layer lattices.

Due to the superior performance of Construction D over Construction A, the construction of convolutional lattices based on Construction D and multi layer Code Lattices are also studied in this section. Construction D relies on two characteristics of the underlying FEC codes: (i) the codes are nested as a chain of sub-codes and (ii) these sub-codes have larger Minimum Euclidean Distance (MED) than the parent code [54, Ch. 8]. Convolutional codes do not readily fulfil such requirements, which may be one reason that convolutional codes have not so far been exploited in Construction D[1]. In this section, we first propose constructing convolutional lattices based on Construction D by neglecting the MED criterion, and then introduce means of increasing the MED of nested convolutional codes by rearranging the input messages which guarantees to fulfil the MED criterion of the Construction D definition.

For the convolutional lattices based on Construction A/D (and also single/multi layer Code Lattices), equivalent encoding based on shift registers is proposed enabling us to exploit existing decoding algorithms of convolutional codes for convolutional lattices, too. The lattice codes based on convolutional lattices allow optimal lattice decoding using the trellis structure of the underlying convolutional code, e.g. the BCJR algorithm. A further contribution of this section is to provide methods to incorporate the BCJR algorithm in lattice decoding. This requires the statistical characteristics of Modulo Lattice Additive Noise (MLAN), and therefore we also derive the probability density function (pdf) of MLAN in closed form for lattices with hypercubic shaping regions. A rather similar pdf has been described in [58, Sec. III-B], however, no closed-form expression for the pdf was derived.

The new lattice decoding algorithms we develop are based on ML/MAP decoders, and thus have similar complexity. However, throughout the section it will be observed that on the point to point channel ML/MAP decoders outperform the corresponding lattice decoders in practice, i.e. in dimensions less than infinity [16, 134–136]. This might raise the question of the benefits of lattice decoding as compared to pure ML/MAP decoding. Our motivation, however, extends beyond the point-to-point channel to relay communication systems [137], and in particular, recently-proposed communication paradigms such as Compute and Forward (CaF), which relies purely on the structure of the lattice, and hence requires practical implementation of lattice decoding for lattices with arbitrarily high dimension, for which direct ML/MAP decoding would be prohibitively complex. Here we study lattice encoding and decoding algorithms in a point to point communication system as a step towards their use in CaF decoders.

This section is organised as follows: In Section 8.2.2 a point-to-point system is introduced. In Sec 8.2.3 the statistical characteristics of MLAN are studied and in Sec 8.2.4 convolutional lattices based on Construction A are proposed, along with the methods for lattice decoding and in Section 8.2.5 a CaF system based on construction A is studied. Section 8.2.6 deals with convolutional lattices based on Construction D and their lattice decoding methods using the trellis structure of the underlying codes. Section 8.2.7 gives concluding remarks, including a discussion of further work required to apply the methods described to CaF, and to turbo lattices.

[1] Although [128, 129] study Turbo lattices based on Construction D, they neglect the minimum distance criterion, and consequently it may result in degradation of the lattice code performance.

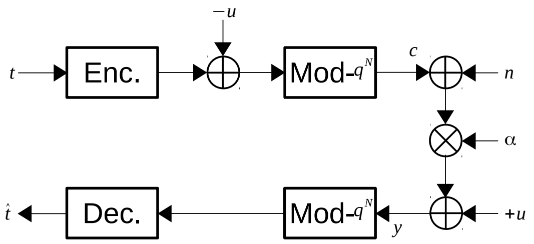


Figure 8.4: System model.

8.2.2 System model

A point-to-point communication system exploiting nested lattice codes according to [16], and illustrated by Figure 8.4, is investigated in this section: the transmitter employs a lattice encoder which maps a message t to a Euclidean codeword c to be sent to the destination, i.e.,

$$c = [t \cdot G_\Lambda - u] \bmod - \Lambda \tag{8.36}$$

where u is a dithering signal, known to the transmitter and receiver, that is uniformly distributed in the Voronoi region of the coarse lattice. G_Λ is the generator matrix of the lattice code that is obtained from a feed forward convolutional encoder according to Construction A and D in this section. Note that the code rate is specified by the shaping lattice, i.e. the number of the lattice points inside the Voronoi region of the shaping lattice as well as the rate of the underlying convolutional code. Assuming M to be the number of lattice points inside the shaping region,

$$R = \frac{1}{N} \log_2 M \tag{8.37}$$

is defined as the code rate where N is the lattice dimension. The signal received at the destination is corrupted by Additive White Gaussian Noise (AWGN) as

$$v = c + n, \tag{8.38}$$

that is multiplied by the α coefficient to implement Minimum Mean Square Error (MMSE) estimation and the dither u is also added to the received signal[2],

$$y = \alpha v + u. \tag{8.39}$$

The signal y is then decoded by a lattice decoder and the transmitted message is recovered at the destination

$$\begin{aligned} \hat{c} &= [\mathcal{Q}(y)] \bmod - \Lambda && (8.40) \\ \hat{t} &= \mathcal{D}(\hat{c}) && (8.41) \end{aligned}$$

[2] Please see [16] for detailed description about the role of MMSE estimator α and the dither u.

where $\mathcal{Q}(\,\cdot\,)$ indicates a lattice quantiser/decoder and $\mathcal{D}$ maps a codeword to a message. Note that the $[\,\cdot\,]\text{mod} - \Lambda$ operation is a distributive operation and so one can rewrite (8.40) as

$$\hat{c} = \mathcal{Q}([y] \bmod - \Lambda) \tag{8.42}$$

which is equivalent to performing the modulo operation before lattice quantisation. Indeed performing the modulo operation before or after lattice decoding/quantisation does not affect the performance of the system and so a common trend in the literature is to apply existing lattice decoding algorithms, e.g. [138, 139], before the modulo operation since this leaves the structure of the lattice intact. However, we take a rather different approach and perform the modulo operation before lattice quantisation as in (8.42).

8.2.3 Statistical characteristics of modulo lattice additive noise

Statistical description of the overall receiver noise plays a key role in the design and theoretical analysis of communication systems; for instance, soft decoding algorithms, e.g. the BCJR algorithm, rely on the distribution of the additive noise in the receiver, which is usually modelled by the Gaussian distribution in conventional communication systems. However, in modulo-lattice channels wherein the receiver employs the modulo $-\Lambda$ operation before channel decoding, the additive noise is no longer Gaussian. Indeed, the additive noise lies inside the fundamental Voronoi region of the coarse lattice, and so, unlike the Gaussian noise, the modulo-Λ Gaussian noise does *not* expand the entire space, i.e. $n_{\text{mod}} \notin (-\infty, +\infty)$. Following Erez *et al.* in [16], we will use the notation of "MLAN" (Modulo-Lattice Additive Noise) in this section.

Considering that we assume a lattice with hypercubic shaping region in this section based on Construction A/D, the $[\,\cdot\,]\text{mod}-\Lambda$ operation for an N dimensional lattice can be performed independently per dimension and so in the rest of this section we concentrate on deriving the statistical description of the noise in a single dimension. As n is normally distributed with zero mean and σ^2 variance, αn is also distributed normally. Moreover, the random variable $(1-\alpha)u$ is distributed uniformly. Consequently, the overall noise is the modulo-Λ of sum of two random variables of which one is distributed normally and the other is distributed uniformly:

$$Z = [\,\underbrace{(1-\alpha)u + \alpha n}_{N'}\,]\text{mod} - \Lambda \tag{8.43}$$

where the pdf of N' is derived as

$$f_{N'}(x) = \frac{1}{4\eta_d}\text{erf}\left(\frac{x-\eta_d}{\sqrt{2}\alpha\sigma}, \frac{x+\eta_d}{\sqrt{2}\alpha\sigma}\right) \tag{8.44}$$

with $\eta_d = d(1-\alpha)$; $\text{erf}(x,y) = \text{erf}(y) - \text{erf}(x)$ is the generalised error function. For a proof of (8.44), see Subsection 8.2.8. Figure 8.5 (black line) illustrates the pdf of N' that was derived in (8.44). A modulo-lattice operation is equivalent to mapping the area

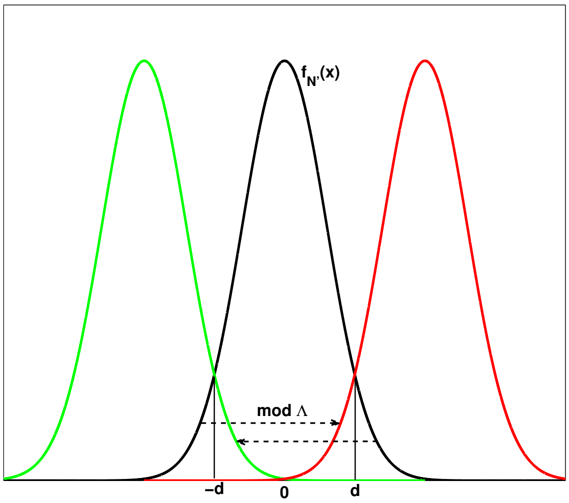


Figure 8.5: A demonstration of $f_{N'}(x) = \frac{1}{4\eta_d}\mathrm{erf}(\frac{x-\eta_d}{\sqrt{2}\alpha\sigma}, \frac{x+\eta_d}{\sqrt{2}\alpha\sigma})$ and modulo operation.

outside Voronoi region into inside the Voronoi region: for instance, in Figure 8.5, the portion of the black curve in $(d, 3d)$ will be mapped inside the $(-d, d)$ region. It is clear from the figure that the portion of $f_{N'}(x)$ in $(d, 3d)$ is equal to the green curve at $(-d, d)$. Note that the green curve corresponds to the pdf of $(1-\alpha)u + \alpha n$ (as in (8.43)) where n follows the same distribution as in (8.43) and u is a random variable uniformly distributed in $(-3d, -d)$. Indeed, the the pdf of modulo-lattice noise in $(-d, d)$ is the sum of an infinite number of random variables with a pdf as in (8.71) with the centres located at $0, \pm 2d, \pm 4d, \cdots$. Consequently, the pdf of Z in (8.43) can be written as

$$f_Z(z) = \frac{1}{4\eta_d}\sum_{i=-\infty}^{\infty}\mathrm{erf}(\frac{z-2di-\eta_d}{\sqrt{2}\alpha\sigma}, \frac{z-2di+\eta_d}{\sqrt{2}\alpha\sigma}). \tag{8.45}$$

Note that $f_Z(z)$ as derived in (8.45) will be used for lattice decoding of convolutional lattices using the BCJR algorithm in the following sections.

Truncation error. Although the expression derived in (8.45) represents $f_Z(z)$ in closed-form, the infinite summation can be considered as a source of inconvenience in practice. Nevertheless, the infinite summation can be truncated with arbitrarily low truncation error. Note that $\lim_{i\to\infty}\mathrm{erf}(\frac{z-2di+\kappa}{\sqrt{2}\alpha\sigma}, \frac{z-2di-\kappa}{\sqrt{2}\alpha\sigma}) = 0$, and so the significance of the expressions in (8.45) decreases as i increases. Figure8.6 illustrates $f_Z(z)$ for various values of σ using the closed-form expression of (8.45) truncated at $i = \pm 2$. The result of Monte Carlo simulations is also provided for comparison: it shows a perfect agreement between the theoretical plot and Monte Carlo simulations even for truncations as low as $i = \pm 2$.

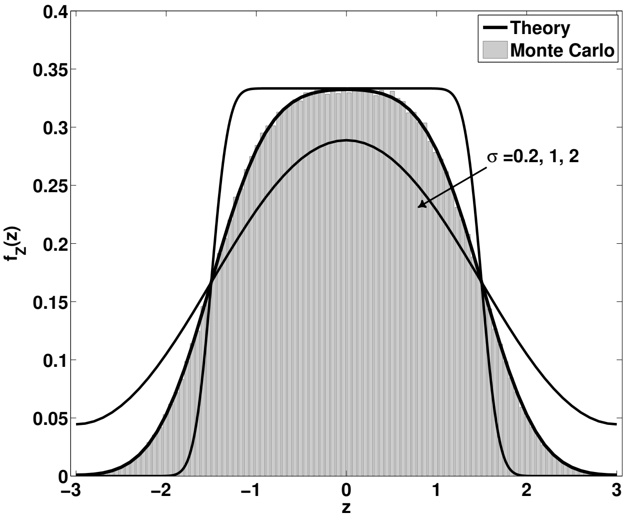


Figure 8.6: pdf of modulo-lattice additive noise with $d = 3$, $\alpha = 0.5$ and various σ. For comparison, Monte Carlo simulations is provided for $\sigma = 1$.

8.2.4 Constructing lattices from forward error correction codes

Lattices with hypercubic Voronoi regions are particularly interesting because the mod-Λ operation in N dimensions can be performed independently in each dimension which results in considerable simplification of the problem. Since the complexity of specifying the Voronoi cell of a non-hypercubic lattice is unbounded in large dimensions [132], in a complexity-performance trade off, hypercubic lattices with lower complexity have been a potential candidate for practical purposes and so we focus on hypercubic lattices, too. “Construction A” and “Construction D” are two well known and widely adopted lattice constructions that have hypercubic Voronoi regions. We adopt them from [54] for constructing convolutional lattices. Moreover, single and multi-layer Code Lattices analogous with Construction A and D, respectively. It will be observed that Code Lattices outperform their counterparts with considerable difference.

Preliminaries: Block convolutional codes

Assuming a k/N block convolutional code; the generator matrix of the block convolutional code is a $k \times N$ matrix where the basis vectors (the rows of a generator matrix) are convolutional codewords generated by setting only one bit of the data vector to one and the rest of the bits to zero. Let us assume that the length-N codeword generated by $[1, 0, \cdots, 0]$ data-word is placed in the first row of the matrix and similarly, the code word generated by a one in the i-th position of the data-word is placed in the i-th row of the generator matrix. Hence, one generator matrix of a $(7,5)$ block convolutional code, according to the above explanation, is

$$G_c = \begin{bmatrix} 1\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0 & \\ 0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,0 & \cdots \\ 0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0 & \\ \vdots & \ddots \end{bmatrix}_{k\times N} . \tag{8.46}$$

In the next sections, (8.46) will be used for lattice construction.

Single layer convolutional lattices: Construction A and code lattice

Construction A. Assume a (k, N, d) linear block code (block convolutional code in this section) in $\mathbb{F}_q$ represented by $\mathcal{C} = \{c_0, c_1, \cdots, c_{M-1}\}$ with generator matrix G_c. Any vector $x = (x_1, \cdots, x_N)$ is a point of an N-dimensional lattice Λ_A, corresponding to codeword $c_i \in \mathcal{C}$ if and only if

$$[x]\text{mod} - q^N \triangleq c_i, \;\; c_i \in \mathcal{C}. \tag{8.47}$$

For instance, assuming $q = 2$, any vector x with even entries, is congruent to the codeword $c_0 = (0, \cdots, 0)$. In other words, any vector x with even entries is (i) a lattice point and (ii) represents the codeword c_0. In the following we discuss how the generator matrix of a convolutional lattice may be obtained from the generator matrix of the convolutional code.

Generator matrix of Λ_A: For a given block convolutional code with rate k/N, the generator matrix of the "N-dimensional" convolutional lattice Λ_A constructed according to Construction A is

$$G_{\Lambda_A} = \begin{bmatrix} G_c \\ G \end{bmatrix}_{N\times N} \tag{8.48}$$

where G_c is the $k \times N$ generator matrix of the convolutional code, e.g. for a $(7,5)$ convolutional code G_c is derived in (8.46) and G is an $(N-k) \times N$ matrix with rows chosen from an $N \times N$ scaled identity matrix $qI_{N\times N}$ wherein the scaling parameter q is specified by the shaping lattice[3]. The matrix G has to be chosen in a way that G_{Λ_A} is a full rank square matrix. Indeed, the role of G is to make G_{Λ_A} a rank N matrix. Minimum Euclidean distance of Λ_A ($d^{\Lambda_A}_{\text{min-u}}$) is

$$d^{\Lambda_A}_{\text{min-u}} = \min\{q, \sqrt{d^{\mathcal{C}}_{\text{min}}}\}. \tag{8.49}$$

where $d^{\mathcal{C}}_{\text{min}}$ is the minimum Hamming weight of the corresponding convolutional code. Note that for $q < \sqrt{d^{\mathcal{C}}_{\text{min}}}$, the error performance of the lattice is expected to be inferior to

[3] In [54], Construction A is described only for $q = 2$, however, it does not necessarily require to be binary; in this section we assume arbitrary q.

the error performance of the underlying block code because the performance of the lattice is bounded by the minimum Euclidean distance, which is smaller than the minimum Euclidean distance of the underlying convolutional code $d^{\Lambda_A}_{\text{min-u}} < \sqrt{d^{\mathcal{C}}_{\text{min}}}$.

Single layer code lattice. The generator matrix of a block convolutional code G_c (size $k \times N$) can also be considered as the generator matrix of a "k-dimensional" lattice, which is called "single layer Code Lattice" ($\Lambda_{\mathcal{C}}$) in this section. Note that although the dimension of the lattice is k, N coordinates are used to represent the lattice points in N dimensional space. The lattice points of $\Lambda_{\mathcal{C}}$ can be generated using $\mathbb{Z} \cdot G_c$ where the size of $\mathbb{Z}$ is $1 \times k$.

Minimum Euclidean distance of $d^{\Lambda_{\mathcal{C}}}_{\text{min-u}}$ is equal to the square of the Hamming distance of the underlying block convolutional code

$$d^{\Lambda_{\mathcal{C}}}_{\text{min-u}} = \sqrt{d^{\mathcal{C}}_{\text{min}}}. \tag{8.50}$$

Remark 1. It can easily be observed that $\Lambda_{\mathcal{C}} \subseteq \Lambda_A$ and so, clearly, $d^{\Lambda_A}_{\text{min-u}} \leq d^{\Lambda_{\mathcal{C}}}_{\text{min-u}}$. Indeed the extension of the $\Lambda_{\mathcal{C}}$ lattice to Λ_A is performed using sub-matrix G in (8.48).

For a better understanding the role of G in (8.48) (or equivalently, the extension of $\Lambda_{\mathcal{C}}$ to Λ_A), a simple two dimensional example is provided in the following: one generator matrix of a two dimensional hexagonal lattice on the $z = 0$ plane is $G_{\text{Hex}} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$, nevertheless, the generator matrix of the lattice is not unique and, for instance, it can be represented by another generator matrix as

$$G_{\text{Hex}} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \tag{8.51}$$

that uses three coordinates on the $x + y + z = 0$ plane to represent a two dimensional lattice. The lattice generated by G_{Hex} in (8.51) is indeed the generator matrix of the two dimensional lattice $\Lambda_{\mathcal{C}}$ discussed above. In order to produce a lattice based on Construction A, one can concatenate, e.g. the row $[0\ 0\ 2]$ with (8.51) and obtain

$$G_{\Lambda_A} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 2 \end{bmatrix}. \tag{8.52}$$

Note that the third row in (8.52) copies the two dimensional hexagonal lattice to the third dimension parallel to $x + y + z = 0$ plane and generates the three dimensional Λ_A lattice. It's worth to mention that, in this example, the $d^{\mathcal{C}}_{\text{min}}$ is the MED of the hexagonal lattice and so $d^{\mathcal{C}}_{\text{min}} = 2$. Consequently $d^{\Lambda_A}_{\text{min-u}} = \min(q, \sqrt{d^{\mathcal{C}}_{\text{min}}}) = \min(2, \sqrt{2}) = \sqrt{2}$ which is bounded by the MED of the code, hence, the error performance of G_{Hex} in (8.51) and G_{Λ_A} in (8.52) are expected to be rather similar; however, for many advanced channel codes, the MED of the code is larger than q and so the error performance of the $\Lambda_{\mathcal{C}}$ lattice is better than the error performance of Λ_A lattice. This will be confirmed by simulation later in this section.

Remark 2. A *transmitter*, exploiting lattices (based on Construction A) as the channel code, generates only the lattice points inside the Voronoi region of the shaping lattice q^N as follows: assuming that $\boldsymbol{u}_{1\times N}$ is the data vector that represents M messages, one can write

$$u_i \in \begin{cases} \{0, 1, \cdots, q-1\}, & \text{for } i \leq k \\ \{0\}, & \text{for } i > k, \end{cases} \tag{8.53}$$

consequently, $\boldsymbol{u} = [\boldsymbol{u}_{\text{data}}\ \boldsymbol{u}_{\text{duplicate}}]$ where $\boldsymbol{u}_{\text{data}}$ is a $1 \times k$ vector that is specified by the first row of (8.53); $\boldsymbol{u}_{\text{duplicate}}$ is a $1 \times (N-k)$ vector that does not carry any information. Note that the transmitter exploits $[\boldsymbol{u} \cdot G_{\Lambda_A}]\text{mod} - q^N$ for assigning a lattice point to a message and so any value assigned to $\boldsymbol{u}_{\text{duplicate}}$ will be discarded by $[\,\cdot\,]\text{mod} - q^N$ operation. Note that one can also generate lattice points generated by $[\boldsymbol{u} \cdot G_{\Lambda_A}]\text{mod} - q^N$ using $[\boldsymbol{u}_{\text{data}} \cdot G_{\Lambda_{\mathcal{C}}}]\text{mod} - q^N$ and so the lattice points inside q^N hypercube are common lattice points between Λ_A and $\Lambda_{\mathcal{C}}$. Note that it is important to distinguish between the two lattices because "*lattice decoding*" using Λ_A or $\Lambda_{\mathcal{C}}$ can lead to different error performance and so one should use appropriate lattice (the lattice with the larger minimum distance) for the purpose of lattice decoding.

Decoding single layer convolutional lattice codes using trellis structure of the code

Although lattice decoding (indeed lattice quantisation) is usually considered to be less complex than ML decoding because of the structure of the lattice, a practical "universal lattice decoding" algorithm with reasonable complexity is still a hot research topic in the field. There are several lattice decoding algorithms proposed in the literature [54, 131, 132, 140], however, the algorithms are only applicable in very low dimensions; for example, in [132] it is clarified that the proposed algorithm has been examined for decoding lattices up to 32 dimensions. Considering that transmission rates close to capacity can be approached only by lattices with high dimension, the existing universal lattice decoding algorithms do not seem to be very appealing in practice.

Apart from the universal lattice decoders, several lattice decoding algorithms have been proposed for certain lattices obtained using particular FEC codes; for instance, lattice decoders based on the sum-product algorithm have been proposed in [18, 141] for Low Density Lattice Codes (LDLC). Likewise, in this section, we are *not* interested in a universal lattice decoding algorithm for an arbitrary lattice but in lattice decoding of convolutional lattices that are obtained using convolutional codes.

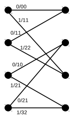


Figure 8.7: Trellis representation of $(7,5)$ convolutional lattice with $q = 3$. Input is in $\{0,1,2\}$ and output is in $\{0,1,2(\text{or } -1)\}$. Green lines represent the transition corresponding to input 0, blue corresponds to input 1 and red for input equal to 2.

Before we continue with lattice decoding of convolutional lattices, let us concentrate on ML/MAP decoding of convolutional lattices and notice that ML and MAP decoding algorithms with reasonable complexity exist (i.e. Viterbi and BCJR, respectively).

ML and MAP decoding of convolutional lattices. For ML/MAP decoding of convolutional lattices, one can simply resort to the trellis structure of the corresponding convolutional code: for instance, assuming $q = 2$ and preserving the order of the basis vectors in the generator matrix of the lattice according to Section 8.2.4-A, the transmitted lattice points are exactly equal to the corresponding binary convolutional code and so one can easily employ the trellis structure of the convolutional code for Viterbi/BCJR decoding of the convolutional lattice. For shaping hypercubes $q > 2$, the trellis structure is not hard to derive. As an example, assume $q = 3$, and so, the lattice encoding is performed in $\mathbb{F}_3$. Assuming a shift register based convolutional encoder which performs operations in $\mathbb{F}_3$, the trellis structure is illustrated by Figure 8.7 for a $(7,5)$ convolutional code where the green arrows represent transitions corresponding to input $u_i = 0$, blue and red arrows show transitions corresponding to $u_i = 1$ and $u_i = 2$, respectively. Note that the overall number of the codewords inside the shaping hypercube is 3^k and so the code rate is

$$R = \frac{1}{n}\log_2 3^k = \frac{1}{2}\log_2 3. \tag{8.54}$$

It is important to clarify that using the trellis structure of the underlying convolutional code for ML/MAP decoding of the lattice, we use the Code Lattice ($\Lambda_{\mathcal{C}}$) for decoding and not the Construction A lattice (Λ_A). Note that $d_{\min}^{\Lambda_{\mathcal{C}}} \geq d_{\min}^{\Lambda_A}$ and consequently decoding on $\Lambda_{\mathcal{C}}$ outperforms decoding on Λ_A.

Lattice decoding of convolutional lattices. As discussed earlier in (8.40) and (8.42), performing the modulo operation before or after lattice quantisation will not affect the performance of the system; therefore, for the purpose of lattice decoding, we perform the modulo operation before lattice quantisation which consequently maps the entire space to the inside of the Voronoi region of the shaping lattice which includes only the lattice points that can actually be transmitted from the source. Therefore, one can exploit the trellis structure of the underlying convolutional lattice for lattice decoding, e.g.

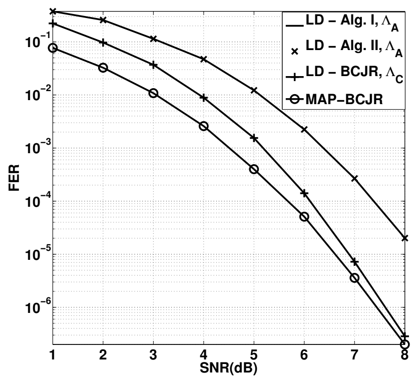


Figure 8.8: Frame Error Rate for $(7,5)$ convolutional lattice with $q=2$.

Viterbi or BCJR. Note that the only difference with the ML/MAP decoding discussed earlier is the $[\cdot]\text{mod} - q^N$ operation and so we refer to this as "Lattice decoding using Viterbi/BCJR" algorithm (where the noise is MLAN) whereas ML/MAP decoding refers to conventional Viterbi/BCJR decoding algorithms without the $[\cdot]\text{mod} - q^N$ operation, i.e. with Gaussian noise. Considering that the BCJR algorithm requires the statistical description of the overall noise, we exploit the pdf of MLAN derived in (8.45) for calculating the state transition probabilities of the BCJR algorithm.

Figure 8.8 illustrates the Frame Error rate (FER) obtained using computer simulations for a $(7,5)$ convolutional lattices with $q=2$, i.e. $\mathbb{F}_2$, on a lattice with dimension equal to 20. Assuming Gaussian noise and modulo operation before BCJR decoding, i.e. Lattice Decoding (LD) using BCJR algorithm, FER is shown by the bold line marked with (+); for comparison two universal lattice decoding algorithms from [130] and [131] are indicated with LD-Algorithm I and II, respectively. It is clear that the proposed BCJR based lattice decoding of convolutional lattices outperforms conventional lattice decoding methods with more than 1.5 dB difference. Note that in Figure 8.8 we are forced to perform computer simulations in low dimension (20 dimension) because existing universal lattice decoders, i.e. LD-Algorithms I and II from [130] and [131], that are used as a benchmark for comparison, are practically feasible only in low dimensions. Figure 8.8 also shows the BCJR decoding of the convolutional lattice (*without* performing modulo operation, say MAP decoding). Note that MAP decoding outperforms lattice decoding because of the poor error performance of lattice decoding[4], however, clearly both the curves converge at high SNR, as expected.

In the following theorem, the advantage of $\Lambda_{\mathcal{C}}$ over Λ_A that leads to a superior error performance as illustrated in Figure 8.8 is discussed.

Theorem 8.2. *Considering error rate as a performance benchmark, lattice decoding of*

[4] Please see [16, 134–136] for a thorough discussion about the error exponent of MLAN channels.

single layer Code Lattice ($\Lambda_{\mathcal{C}}$) outperforms lattice decoding of the corresponds Construction A lattice (Λ_A).

Proof. Comparing the lattice decoders in Figure 8.8 puts forward the question of why the proposed lattice decoder outperforms existing universal lattice decoders? There are indeed two main reasons for this that are explained in the following:

- The proposed lattice decoding using the BCJR algorithm is performed over the Code Lattice ($\Lambda_{\mathcal{C}}$) whereas lattice decoding using universal lattice decoders proposed by [130, 131] is performed over Λ_A. The minimum Euclidean distance of $\Lambda_{\mathcal{C}}$ is equal to the minimum Euclidean distance of the underlying convolutional code, i.e. $d_{\min}^{\Lambda_{\mathcal{C}}} = \sqrt{d_{\min}^{\mathcal{C}}}$, while $d_{\min}^{\Lambda_A} \leq \sqrt{d_{\min}^{\mathcal{C}}}$. Hence, lattice decoding on $\Lambda_{\mathcal{C}}$ using the BCJR algorithm outperforms existing universal lattice decoders that perform decoding on Λ_A lattice. Note that universal lattice decoding algorithms proposed, e.g. [130–132] require a generator matrix in square form and so we are forced to use the Λ_A lattice with square generator matrix for decoding[5].
- In a Λ_A lattice which is generated from a $k \times N$ convolutional code, the dimension of the lattice increases from k to N (using G sub-matrix in (8.48)) and consequently the number of adjacent lattice points that can erroneously be decoded increases.

□

As discussed in earlier sections, considering that a lattice decoder is, in general, outperformed by an ML decoder, application of lattice decoders in practice does not seem to be a justifiable choice in point-to-point communication systems, however, the idea of obtaining convolutional lattices, in this section, was initially motivated by CaF relaying [22]. In order to validate the usefulness of lattice decoding, the next section, we consider a CaF relaying where lattice decoding is the method of choice due to its manageable complexity in practical systems.

8.2.5 Application of convolutional lattices in compute and forward

In this section, we use CaF relaying to validate the usefulness of the proposed lattice decoder by comparing the complexity of the proposed lattice decoder with an ML/MAP decoder. Assume a CaF relaying systems wherein multiple source nodes transmit their data simultaneously towards a relay node. For instance, Figure 8.9 illustrates the Multiple Access Channel (MAC) phase of a relaying system wherein the source nodes employ convolutional lattice codes as the FEC code. The relay node performs channel decoding (whether ML or lattice decoding); upon decoding the resultant lattice point in the relay,

[5] We are not aware of any "universal lattice decoding" algorithm with reasonable decoding complexity which performs on non-square generator matrix. However, if one proposes a universal lattice decoder which performs decoding on a non-square generator matrix ($k \times N$ in this section), we conjecture that the performance should be equivalent to the proposed BCJR (or Viterbi) based lattice decoding.

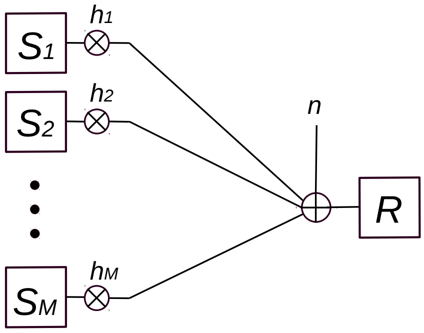


Figure 8.9: System model: Compute and forward.

$[\cdot]\mathrm{mod}-q^N$ operation is performed as network coding and a new lattice point is then sent to the intended destination nodes, see [22] for detailed description about CaF. Assuming the source nodes to apply the same convolutional codes, the complexity of the rely node employing the proposed lattice decoder is considerably lower than the complexity of the equivalent ML/MAP decoder. Let us define the complexity of the proposed lattice decoder for convolutional lattices as the number of trellis states

$$C = q^T \quad \text{trellis states} \tag{8.55}$$

where C is the measure of complexity and $T+1$ is the constraint length of corresponding convolutional code; note that the complexity of the lattice decoder is independent of the number of the source nodes whereas assuming M source nodes, the complexity of ML/MAP decoder is $C = q^{MT}$ trellis states. Consequently, as illustrated by Figure 8.10, the complexity of the ML decoder is indeed much more than the lattice decoder.

Assuming Gaussian channel ($h_i = 1$) and convolutional lattice based on $(7,5)$ code, Figure 8.11 shows the FER of CaF system with two source nodes exploiting convolutional lattices over a hypercube of 3^{10} and 3^{200}, i.e. shaping lattice of $q = 3$ in 10 and 200 dimensions with rate $\frac{1}{2}\log_2 3$, with MAP and proposed lattice decoding used in the relay node: from a complexity point of view, the proposed lattice decoder using BCJR algorithm performs decoding with only $3^2 = 9$ trellis states whereas an ML/MAP decoder has $3^{2\times 2} = 81$ trellis states. This demonstrates the advantage of the proposed lattice decoder in certain communication systems like CaF. Note that since the complexity of an ML/MAP decoder grows exponentially with the number of the users, practical implementation of it is indeed impossible with moderate and large number of users. Note that as expected, ML/MAP decoder has a better performance but this is true only at low SNR and the performance of lattice and ML/MAP decoders converge at high SNR, as illustrated by Figure 8.11; nevertheless, taking complexity into account for a CaF system with large number of the users, the proposed lattice decoder will be the method of choice.

Further focusing on Figure 8.11 reveals that the gap between MAP and lattice decoders increases with increasing dimension of the lattice and so the two curves converge at

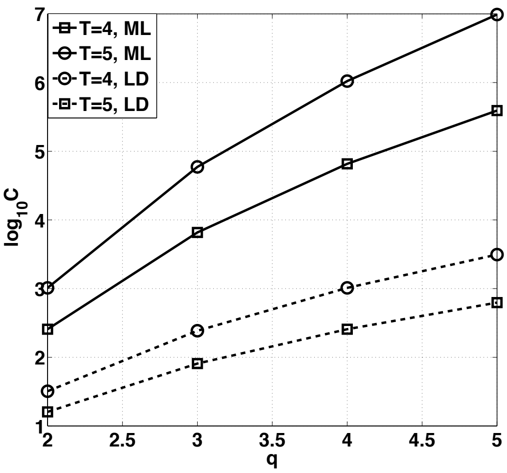


Figure 8.10: Complexity of ML and lattice decoding for various values of T and $M = 5$.

higher SNRs. For instance, assuming a 10 dimensional convolutional lattice, the two curves have almost converged at $9\,\mathrm{dB}$ whereas for the convolutional lattice with 200 dimensions, there is a rather large gap between the two curves at $9\,\mathrm{dB}$. The reason for this is explained in the following *Remark*. Note that we do not provide comparison with universal lattice decoders from e.g. [130] and [131] due to the large dimension of the lattice, however, it is clear, from Figure 8.8, that the proposed lattice decoding outperforms them.

Remark 3. There is one major concern to be discussed here: why does the gap between lattice decoder and ML/MAP decoder increase with increasing dimension of the lattice? Indeed by increasing the dimension of a lattice the number of edge points[6] that can be decoded erroneously increases; this is the main reason that the 200 dimensional lattice in Figure 8.11 converges at higher SNR in comparison with the equivalent 10 dimensional lattice. Moreover, as discussed earlier, due to the implementation complexity of universal lattice decoders in higher dimensions, no comparison is provided with universal lattice decoders, however, we conjecture that the gap will be even wider for universal lattice decoders because the dimension of the Λ_A lattice (for the $(7,5)$ convolutional code used in Figure 8.11) is twice that of $\Lambda_{\mathcal{C}}$ and so it results in further degradation of the performance. Nevertheless, we expect that as $k \to \infty$, the gap to be closed and both the ML and LD decoding curves converge.

[6] By "edge points" we mean the lattice points in the boundary of the shaping lattice that can be transmitted by the transmitter and can erroneously be decoded to the lattice points outside the shaping lattice.

8.2.6 Encoding and decoding multilayer convolutional lattices

Lattices based on Construction D (Λ_D) and what we refer to as "multilayer code lattices ($\Lambda_{\mathcal{C}}$)[7]" are the two lattice constructions discussed in this section. However, before we continue with the definition of Construction D and the construction of lattices from convolutional codes exploiting the Construction D template, we would like to focus also on conventional multilevel coding [142] techniques which are referred to as Construction by Code Formula (CCF). This is discussed in the next subsection. Note that we are interested in CCF because both CCF and Construction D are usually regarded equivalent in the literature (see [143–145]).

Multilevel codes or construction by code formula

Assume a family of a binary linear codes in which

$$\mathbb{F}_2^N \supseteq \mathcal{C}_1 \supseteq \mathcal{C}_2 \cdots \supseteq \mathcal{C}_a \tag{8.56}$$

with $\mathcal{C}_i$ as a $[k_i, N, d_i]$ linear block code. A code based on CCF will be defined as

$$\mathcal{C}_{\mathrm{CCF}} = \psi_1(\mathcal{C}_1) + \psi_2(\mathcal{C}_2) + \cdots + \psi_a(\mathcal{C}_a) \tag{8.57}$$

where $\psi(\,\cdot\,)$ is a map from $\mathbb{F}_2^N$ to $\mathbb{R}^N$ in which $\psi_i(\boldsymbol{x}) = \frac{\boldsymbol{x}}{2^{i-1}}$ where $\boldsymbol{x} \in \mathcal{C}_i$. For instance, assuming $a = 3$, one can write $\mathcal{C}_{\mathrm{CCF}} = \frac{1}{4}\mathcal{C}_3 + \frac{1}{2}\mathcal{C}_2 + \mathcal{C}_1$ for the resultant code[8]. The code rate R_{CCF} of CCF is $R_{\mathrm{CCF}} = \sum_{i=1}^{a} R_{\mathcal{C}_i}$. The desired aspects of CCF, among the others, is that encoding (and decoding) $\mathcal{C}_{\mathrm{CCF}}$ can be performed using the conventional encoding (and decoding) methods used for the underlying code $\mathcal{C}_i$. For instance, one can use Viterbi or BCJR algorithm for decoding a convolutional $\mathcal{C}_{\mathrm{CCF}}$, wherein the receiver decodes the inner layer $\mathcal{C}_a$ first and exploits it as *a priori* knowledge passed to the decoder which decodes the layer corresponding to $\mathcal{C}_{a-1}$. This multi-stage decoding algorithm continues until all the layers are decoded. Note that multilevel codes and multi-stage decoding algorithms are extensively studied in the literature (see [142]) and so we will adopt them in the following for encoding and decoding lattices based on Construction D.

In the following, taken from [54], we will define Construction D and will explain, using a counterexample, that Construction D and CCF can be different.

Construction D

Assume a family of a binary linear codes in which

$$\mathbb{F}_2^N \supseteq \mathcal{C}_1 \supseteq \mathcal{C}_2 \cdots \supseteq \mathcal{C}_a \tag{8.58}$$

[7] That is analogous to single layer code lattice discussed in Section 8.2.4.

[8] Alternatively, one can write $\mathcal{C}_{\mathrm{CCF}} = \mathcal{C}_3 + 2\mathcal{C}_2 + 4\mathcal{C}_1$, too.

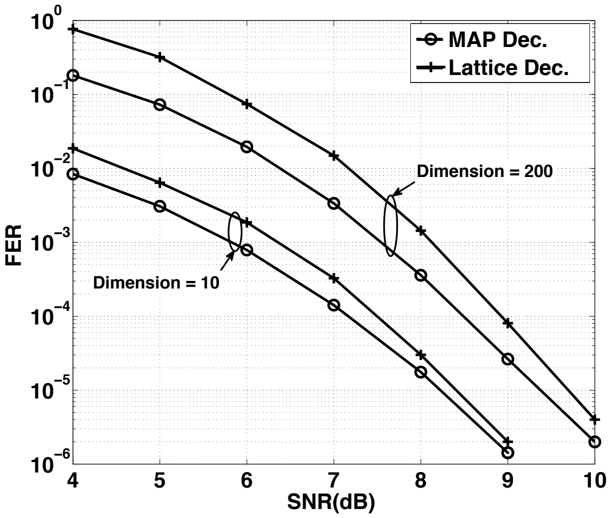


Figure 8.11: Frame error rate for $(7,5)$ convolutional lattice with $q = 3$.

with $\mathcal{C}_i$ as a $[k_i, N, d_i]$ linear block code where $d_i \geq 4^i/\gamma$ in which $\gamma = 2$ or 4. Choose N basis vectors of $\mathbb{F}_2^N$, i.e. $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_N\}$, wherein the set of $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{k_i}\}$ spans $\mathcal{C}_i$. Also assume that $\psi(\,\cdot\,)$ is a map from $\mathbb{F}_2^N$ to $\mathbb{R}^N$ in which $\psi_i(\boldsymbol{x}) = \frac{\boldsymbol{x}}{2^{i-1}}$ where $\boldsymbol{x} \in \mathcal{C}_i$. A lattice based on Construction D contains all vectors of the form

$$\Lambda_D = \sum_{i=1}^{a} \sum_{j=1}^{k_i} \alpha_j^{(i)} \psi_i(\boldsymbol{b}_j) \; + \; (2\mathbb{Z})^N \tag{8.59}$$

where $\alpha_k^{(j)} \in \{0,1\}$. Consequently, Λ_D is an N dimensional lattice with hypercube fundamental Voronoi region. Assuming $\mathcal{C}_D$ to be a code consisting of all lattice points inside the $(-1,1]^N$ hypercube of Λ_D lattice, any point/vector $\boldsymbol{x} = (x_1, \cdots, x_N) \in \mathbb{R}^N$ is a lattice point congruent to codeword c_i if and only if

$$[x]\text{mod} - 2^N \triangleq c_i, \;\; c_i \in \mathcal{C}_D. \tag{8.60}$$

Note that as both the CCF and Construction D are defined based on the $\psi(\,\cdot\,)$ function, they are usually considered to be equivalent in the literature, e.g. [143–145]), however, in order to disprove this conjecture, a counterexample is provided in the following which shows that CCF does not necessarily result in a lattice construction.

Counterexample: Assume two nested binary codes $\mathcal{C}_1 \supseteq \mathcal{C}_2$ with $G_1 = \begin{bmatrix} 1100 \\ 1010 \\ 1001 \end{bmatrix}$ and $G_2 = \begin{bmatrix} 1100 \\ 1010 \end{bmatrix}$. The codebook of CCF inside the 2^4 hypercube obtained from $\mathcal{C}_{\text{CCF}} =$

$\frac{1}{2}\mathcal{C}_2 + \mathcal{C}_1$ is

$$\begin{aligned} \mathcal{C}_{\mathrm{CCF}} = \Big\{ & [0\ 0\ 0\ 0]\ [0\ 0\ 1\ 1]\ [0\ 1\ 0\ 1]\ [0\ 1\ 1\ 0] \\ & [1\ 0\ 0\ 1]\ [1\ 0\ 1\ 0]\ [1\ 1\ 0\ 0]\ [1\ 1\ 1\ 1] \\ & [0.5\ 0\ 0.5\ 0]\,,[0.5\ 0\ 1.5\ 1]\,,[0.5\ 0.5\ 0\ 0]\,,[0.5\ 0.5\ 1\ 1] \\ & [0.5\ 1\ 0.5\ 1]\,,[0.5\ 1\ 1.5\ 0]\,,[0.5\ 1.5\ 0\ 1]\,,[0.5\ 1.5\ 1\ 0] \\ & [1.5\ 0\ 0.5\ 1]\,,[1.5\ 0\ 1.5\ 0]\,,[1.5\ 0.5\ 0\ 1]\,,[1.5\ 0.5\ 1\ 0] \\ & [1.5\ 1\ 0.5\ 0]\,,[1.5\ 1\ 1.5\ 1]\,,[1.5\ 1.5\ 0\ 0]\,,[1.5\ 1.5\ 1\ 1] \\ & [0\ 0.5\ 0.5\ 0]\,,[0\ 0.5\ 1.5\ 1]\,,[0\ 1.5\ 0.5\ 1]\,,[0\ 1.5\ 1.5\ 0] \\ & [1\ 0.5\ 0.5\ 1]\,,[1\ 0.5\ 1.5\ 0]\,,[1\ 1.5\ 0.5\ 0]\,,[1\ 1.5\ 1.5\ 1] \Big\}. \end{aligned} \tag{8.61}$$

The generator matrix of Construction D is[9]

$$G_{\Lambda_D} = \begin{bmatrix} \frac{1}{2}\frac{1}{2}00 \\ \frac{1}{2}0\frac{1}{2}0 \\ 1001 \\ 2000 \end{bmatrix}, \tag{8.62}$$

with which, the lattice points inside the 2^4 hypercube are

$$\begin{aligned} \mathcal{C}_D = \Big\{ & [0\ 0\ 0\ 0]\ [0\ 0\ 1\ 1]\ [0\ 1\ 0\ 1]\ [0\ 1\ 1\ 0] \\ & [1\ 0\ 0\ 1]\ [1\ 0\ 1\ 0]\ [1\ 1\ 0\ 0]\ [1\ 1\ 1\ 1] \\ & [0.5\ 0\ 0.5\ 0]\ [0.5\ 0\ 1.5\ 1]\ [0.5\ 0.5\ 0\ 0]\ [0.5\ 0.5\ 1\ 1] \\ & [0.5\ 1\ 0.5\ 1]\ [0.5\ 1\ 1.5\ 0]\ [0.5\ 1.5\ 0\ 1]\ [0.5\ 1.5\ 1\ 0] \\ & [1.5\ 0\ 0.5\ 1]\ [1.5\ 0\ 1.5\ 0]\ [1.5\ 0.5\ 0\ 1]\ [1.5\ 0.5\ 1\ 0] \\ & [1.5\ 1\ 0.5\ 0]\ [1.5\ 1\ 1.5\ 1]\ [1.5\ 1.5\ 0\ 0]\ [1.5\ 1.5\ 1\ 1] \\ & [0\ 0.5\ 0.5\ 1]\ [0\ 0.5\ 1.5\ 0]\ [0\ 1.5\ 0.5\ 0]\ [0\ 1.5\ 1.5\ 1] \\ & [1\ 0.5\ 0.5\ 0]\ [1\ 0.5\ 1.5\ 1]\ [1\ 1.5\ 0.5\ 1]\ [1\ 1.5\ 1.5\ 0] \Big\}. \end{aligned} \tag{8.63}$$

Careful comparison of (8.61) and (8.63) reveals that the two last rows in (8.61) and (8.63) are different and so, one can easily conclude that CCF and Construction D are not *necessarily* equivalent. Furthermore, although $[1.5\ 1.5\ 1\ 1]$ and $[0\ 1.5\ 1.5\ 0]$ are points of $\mathcal{C}_{\mathrm{CCF}}$ in (8.61) (the last vectors in row six and seven), their mod-2 sum

$$\big[[1.5\ 1.5\ 1\ 1] + [0\ 1.5\ 1.5\ 0]\big]\mathrm{mod} - 2^4 = [1.5\ 1\ 0.5\ 1] \tag{8.64}$$

does not belong to $\mathcal{C}_{\mathrm{CCF}}$ in (8.61) and so, clearly, in this example, the CCF does not generate lattice points. In general, the CCF does not necessarily generate a lattice, however,

[9] Note that obtaining generator matrix of a lattice based on Construction D will be explained in further detail in the following, however, in order to validate the difference between Construction D and CCF, we use it in this example without a proof.

for the particular case of convolutional codes, we introduce an approach with which lattice points of Construction D are generated using CCF. ■

Note that on the one hand we are interested in constructing a lattice from (convolutional) codes based on Construction D and, on the other hand, we would like to make Construction D and CCF equivalent because then we can exploit existing and practically feasible decoding algorithms of CCF for decoding convolutional lattices that are constructed based on Construction D, else, as discussed in the context of construction A, *universal lattice decoders* are not interesting from a practical point of view at high dimensions.

In the following, we focus on deriving the generator matrix of a lattice constructed from convolutional codes according to Construction D and its equivalent CCF. For this, we first neglect the minimum distance criterion of Construction D definition in the following, i.e. the $d_i \geq 4^i/\gamma$ criterion. Later on, we will discuss methods for ensuring the minimum distance criterion is fulfilled which indeed can improve the performance of the code[10].

No minimum distance criterion. The generator matrix of the convolutional lattice will be obtained as follows: the first k_a basis vectors[11] multiplied by $1/2^{a-1}$ form the first k_a rows of the generator matrix; the rows from k_a+1 to k_{a-1} are obtained by multiplying the k_a+1 to k_{a-1} rows of the convolutional code generator matrix G_c to $1/2^{a-2}$. Similarly one can obtain all the rows of the generator matrix of the lattice using the $\psi(\,\cdot\,)$ mapping that corresponds to the associated convolutional code. The remaining $(N-k)$ rows are chosen from the rows of a $2I_{N\times N}$ matrix in such a way that the generator matrix of the lattice has rank N.

In the following, the generator matrix of a convolutional lattice based on Construction D is further discussed using an example.

Example 1: Assume three $(7,5)$ convolutional codes as $\mathbb{F}_2^N \supseteq \mathcal{C}_1 \supseteq \mathcal{C}_2 \supseteq \mathcal{C}_3$ and let $G_{\mathcal{C}_i}$ be the generator matrix of a $(7,5)$ convolutional code as derived in (8.46). Note that since we assume nested codes, we mean that all $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$ contain equal length codewords (and data words) and so it implies that the data words of the sub-codes are zero padded (ZP) to make the length of the data vectors of the sub-codes equal to the length of the data vector of the parent code $\mathcal{C}_1$. For instance assume $k_3 = 1$, $k_2 = 2$, $k_1 = 4$; therefore, the generator matrix of the the three nested codes is as follows: $G_{\mathcal{C}_3} = [\boldsymbol{b}_1]_{1\times 8}$,

[10] Note that the main reason for neglecting the minimum distance criterion is due to a lack of *nested* convolutional codes that fulfil the minimum distance criterion. Therefore, many papers, e.g. [128, 129, 146], relax this criterion.

[11] The basis vectors are obtained according to the description in Section 8.2.4.

$G_{\mathcal{C}_2} = [\boldsymbol{b}_1; \boldsymbol{b}_2]_{2\times 8}$, $G_{\mathcal{C}_1} = [\boldsymbol{b}_1; \boldsymbol{b}_2; \boldsymbol{b}_3; \boldsymbol{b}_4]_{4\times 8}$ where $\boldsymbol{b}_i$ is the i-th row of (8.46). The generator matrix of the lattice constructed according to Construction D is

$$G_{\Lambda_D} = \begin{bmatrix} \frac{1}{4}\boldsymbol{b}_1 \\ \frac{1}{2}\boldsymbol{b}_2 \\ 1\boldsymbol{b}_3 \\ 1\boldsymbol{b}_4 \\ G \end{bmatrix}_{8\times 8} . \tag{8.65}$$

Note that the coefficient $\frac{1}{4}$ for $\boldsymbol{b}_1$, the coefficient $\frac{1}{2}$ for $\boldsymbol{b}_2$ and 1 for $\boldsymbol{b}_3$ and $\boldsymbol{b}_4$ represent the $\psi_i(\,\cdot\,)$ function in (8.59) and G contributes the $(2\mathbb{Z})^N$ part of (8.59). All the lattice points inside the $(-1, 1]^N$ hypercube are the codewords of the Construction D convolutional lattice, and are obtained using the $[\boldsymbol{u} \cdot G_{\Lambda_D}] \bmod - 2^N$ operation where $\boldsymbol{u} \in \mathbb{Z}$.

In order to take advantage of decoding algorithms of CCF, in the following, a method is introduced by which Construction D and CCF are equivalent.

Construction D using CCF: In the above example, there are three nested codes, contributing in encoding four bits (say $\{d_1, d_2, d_3, d_4\}$ corresponding to basis vectors $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_4\}$, respectively). Consequently, one can say that d_1 is encoded by $\mathcal{C}_3$, d_2 by $\mathcal{C}_2$ and d_3, d_4 by $\mathcal{C}_1$. In order to generate Construction D lattice points inside the 2^4 hypercube, similar to multilevel codes (or CCF), one can write the equivalent transmitter side generator matrix as

$$G^{\text{TX-eqv}}_{\Lambda_D} = \begin{bmatrix} \begin{bmatrix} \frac{1}{4}\boldsymbol{b}_1 \end{bmatrix} \\ \begin{bmatrix} \frac{1}{2}\boldsymbol{b}_1 \\ \frac{1}{2}\boldsymbol{b}_2 \end{bmatrix} \\ \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \end{bmatrix} \end{bmatrix} , \tag{8.66}$$

however, set the data bits corresponding to the basis vectors indicated by the red color to zero, i.e.

$$\boldsymbol{d}_{\text{eq}} = [\underbrace{\,d_1\,}_{\mathcal{C}_3}, \underbrace{0, d_2}_{\mathcal{C}_2}, \underbrace{0, 0, d_3, d_4}_{\mathcal{C}_1}]. \tag{8.67}$$

Consequently, $\left[\boldsymbol{d}_{\text{eq}} \cdot G^{\text{TX-eq}}_{\Lambda_D}\right] \bmod - 2$ will generate the same lattice points inside the 2^4 hypercube that will be generated by $[\boldsymbol{d} \cdot G_{\Lambda_D}] \bmod - 2^4$ where $\boldsymbol{d} = [d_1, d_2, d_3, d_4]$.

Note that the upper sub-matrix in (8.66) corresponds to $\mathcal{C}_3$, the sub-matrix in the middle corresponds to $\mathcal{C}_2$ and the bottom one corresponds to $\mathcal{C}_1$; hence, instead of using matrix multiplication for generating lattice points (codewords), one can use conventional shift register based encoders according to Figure 8.12 in the transmitter where

$$\boldsymbol{d}_3 = [d_1, \underbrace{0, 0, 0}_{\text{ZP}}] \tag{8.68a}$$

$$\boldsymbol{d}_2 = [0, d_2, \underbrace{0, 0}_{\text{ZP}}] \tag{8.68b}$$

$$\boldsymbol{d}_1 = [0, 0, d_3, d_4]. \tag{8.68c}$$

Consequently, $[\boldsymbol{d} \cdot G_{\Lambda_D}] \bmod -2$, $\left[\boldsymbol{d}_{\text{eq}} \cdot G_{\Lambda_D}^{\text{TX-eq}}\right] \bmod -2$ and Figure 8.12 will generate the same lattice points inside the 2^4 hypercube.

Ensuring minimum distance criterion. So far, we have relaxed the minimum distance criterion of the original definition of Construction D. The relaxation of the minimum distance criterion was, in part, to focus on constructing lattices using conventional convolutional codes; however, the literature usually ignores the minimum distance criterion, e.g. [128, 141, 146], since it is hard to find nested codes that fulfil this criterion. In particular, in the case of convolutional codes, the minimum distance of a code is fixed; for instance, it is well known that the minimum distance of the $(7, 5)$ convolutional code that was used in the above example is equal to five.

In the previous part, we applied a naive method of zero padding to have data words of equal length in all convolutional codes (see (8.68)), however, one can perform repetition of the uncoded data bits instead of simply zero padding which indeed can result in increasing the minimum distance of the code.

For clarity of explanation, let us begin with the same $(7, 5)$ convolutional code with the generator matrix derived in (8.46). Note that (8.46) is only one of the generator matrices of the $(7, 5)$ code; each basis vector in (8.46) can be replaced by another basis vector. For instance, the basis vector in the first row of (8.46), i.e. $\boldsymbol{b}_1 = [11101100\cdots]$, can be replaced by a new $\boldsymbol{b}_1$ where $\boldsymbol{b}_1 = [110101110\cdots]$ that is a codeword generated by a data vector with the first two bits set to 1 and the rest of the bits set to zero. Note that in *Example 1* where only one bit d_1 is transmitted by the inner code $\mathcal{C}_3$, one can repeat d_1 instead of zero padding: indeed, substituting $\boldsymbol{b}_1 = [11101100\cdots]$ with $\boldsymbol{b}_1 = [110101110\cdots]$ in (8.65) is equivalent to transmitting $[d_1, d_1, 0, 0]$ from $\mathcal{C}_3$. Moreover, considering that $\mathcal{C}_3$ produces only two codewords (all zero and $[110101110000]$), clearly, the minimum distance of the code has increased to 6. Note that codes with larger minimum distance can be produced by different repeating patterns for different convolutional codes: for instance, repeating d_1 according to $[d_1, 0, 0, d_1]$ is equivalent to replacing $\boldsymbol{b}_1 = [11101100\cdots]$ with $\boldsymbol{b}_1 = [11101111101100]$ and so $\mathcal{C}_3$ will produce two codewords (all zero and $[11101111101100]$) that have minimum distance equal to 10.

Note that the repeating pattern depends on the convolutional code, however, for the particular example of the $(7, 5)$ convolutional code, repeating $[d_i00]$ along the data word, *instead of zero padding*, will generate codewords with maximum MED.

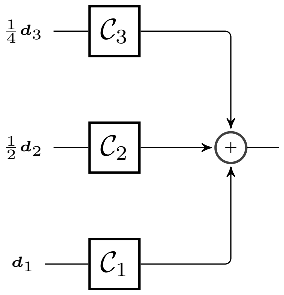


Figure 8.12: Equivalent Construction D encoder using conventional FEC encoders.

Decoding multilayer convolutional lattice codes

One obvious way of lattice decoding convolutional lattices obtained using Construction D is to employ the well known universal lattice decoders in combination with the original generator matrix of Λ_D as obtained, e.g. in (8.65). However, as observed in Section 8.2.4, the performance will be poor due to the edge lattice points and the MED of the lattice which is upper bounded by 2. Another solution is to decode the lattice in k dimensions instead of N and obtain better performance; this is discussed in the following. The following corollary is provided as a result of Theorem 8.2 for multilayer convolutional lattices:

Corollary 8.1. *Considering error rate as a performance benchmark, lattice decoding of multilayer Code Lattice ($\Lambda_{\mathcal{C}}$) outperforms lattice decoding of the corresponds Construction D lattice (Λ_D).*

Proof. The proof of single layer convolutional lattices in Theorem 8.2 proves the corollary, too. □

Note that, in the following, lattice decoding is performed over $\Lambda_{\mathcal{C}}$ (rather than Λ_D). However, we use the term "Construction D" to refer to multilayer convolutional lattices. Considering that lattice codes based on Construction D, as described in the previous subsection, are indeed multilevel convolutional codes [142] and so one can apply multi-stage trellis decoding algorithms for ML, MAP or lattice decoding (similar to Construction A in Section 8.2.4). The multi-stage decoding is started by decoding the inner code with the largest minimum distance $\mathcal{C}_a$; the decoded layer is then fed to the higher layer $\mathcal{C}_{a-1}$ and is used as *a priori* information for decoding the data of the corresponding layer. The process is continued to until decoding $\mathcal{C}_1$. Note that any known decoding algorithms, e.g. Viterbi or BCJR, can be applied for decoding the layers (we are interested in BCJR decoding in this section as it is a SISO decoder). The BCJR decoder for multi-stage decoding is slightly different than the Conventional BCJR decoder because the state transition probabilities depend on the *a priori* information of the other layers, too. *A priori* information

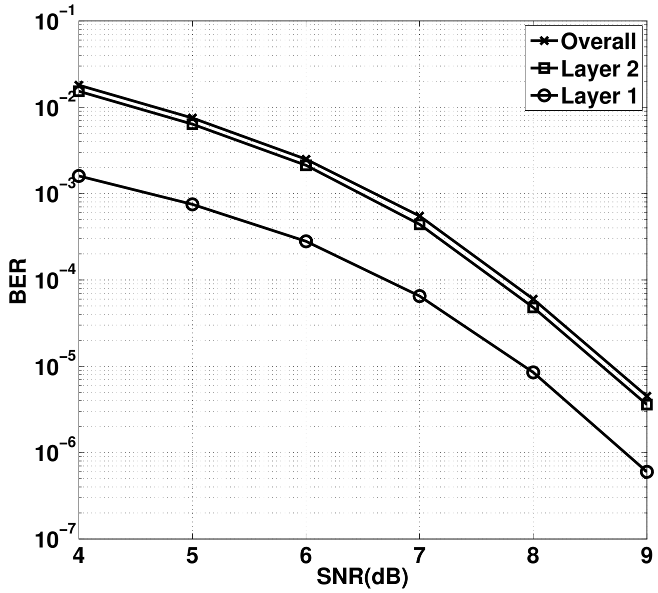


Figure 8.13: BER – Construction D.

can be hard information plugged from inner layers to upper layers or soft information that only passes the probability of the data corresponding to the other layers; by exchanging soft information among the layers, iterative decoding of the layers is also possible that indeed offers higher performance gains.

Example 2: A design methodology of a two layer lattice based on construction D from $(7,5)$ convolutional codes is explained in this example. Assume that the nested codes are defined as $\mathbb{F}_2^N \supseteq \mathcal{C}_1 \supseteq \mathcal{C}_2$ where the dimension $N = 300$, $\mathcal{C}_1$ and $\mathcal{C}_2$ together participate in encoding 150 bits, i.e. number of messages $M = 2^{150}$. Moreover, assume $\mathcal{C}_2$ participates in encoding 10 bits and $\mathcal{C}_1$ encodes 140 bits. In the following we will describe obtaining the generator matrix of the lattice based on Construction D where the minimum Euclidean distance criterion of the Construction D definition is fulfilled. Later on, a corresponding lattice encoder based on conventional convolutional encoders is described.

Lattice generator matrix

One lattice based on construction D will be obtained as follows:

- *Inner layer:* the inner code $\mathcal{C}_2$ carries only 10 data bits in inner layer of a codeword of length 300 that is generated from a data word of length 150 (because it is based on a $(7,5)$ convolutional code); consequently, each data bit in inner layer can be represented by 15 “virtual” data bits that can be arranged in the desired

arrangement in order to achieve the desired MED. As there are 10 data bits to be encoded in the inner layer, and there are only 10 basis vectors that need to be specified; we propose $\boldsymbol{b}_1 = \bigoplus r_i$ where $i \in \{1, 4, 7, 10, 13\}$ and r_i specifies the rows of the generator matrix of $(7, 5)$ code that was derived in (8.46); $\bigoplus$ represents mod-2 summation. In other words, $\boldsymbol{b}_1$ is a codeword generated by a data word, in octal notation $\boldsymbol{d}_{\boldsymbol{b}_1} = [4, 4, 4, 4, 4, \mathbf{0}_{1\times135}]$. Hence, feeding $\boldsymbol{d}_{\boldsymbol{b}_1}$ to a $(7, 5)$ convolutional encoder generates the first basis vector.
Likewise, $\boldsymbol{b}_2 = \bigoplus r_i$ where $i \in \{16, 19, 22, 25, 28\}$ that is generated from a data word in octal notation as $\boldsymbol{d}_{\boldsymbol{b}_2} = [\mathbf{0}_{1\times15}, 4, 4, 4, 4, 4, \mathbf{0}_{1\times120}]$. One can similarly obtain all the basis vectors of the inner generator matrix, e.g. the 10-th basis vector is $\boldsymbol{b}_{10} = \bigoplus r_i$ with $i \in \{136, 139, 142, 145, 148\}$ that is generated from the $\boldsymbol{d}_{\boldsymbol{b}_{10}} = [\mathbf{0}_{1\times135}, 4, 4, 4, 4, 4]$ data vector. Consequently, plugging in $\boldsymbol{d}_{\boldsymbol{b}_1 0}$ to a $(7, 5)$ convolutional encoder will generate the basis vectors corresponding to the inner layer, i.e. the first ten rows of Λ_D.

- *Outer layer:* One can arbitrarily choose the 140 remaining basis vectors, i.e.

$$\boldsymbol{b}_{11}, \boldsymbol{b}_{12}, \cdots, \boldsymbol{b}_{150}$$

from the generator matrix (8.46) of the convolutional code; however, the only constraint is that for any $i \geq 11$, the summation $\oplus\, \boldsymbol{b}_i \neq \boldsymbol{b}_k$ where $k = 1, 2, \cdots, 10$. The constraint is stressed to make sure that the generator matrix of the lattice is full rank. The remaining 150 basis vectors are chosen from $2\boldsymbol{I}_{300\times300}$ matrix with which G_{Λ_D} is a full rank matrix.

Construction D using CCF. The equivalent shift register based encoder consists of two conventional convolutional encoders where 140 data bits are encoded by $\mathcal{C}_1$ and 10 data bits by $\mathcal{C}_2$; moreover, the data of $\mathcal{C}_2$ are repeated in the corresponding positions to generate a "virtual" data word of length 150. Figure 8.13 illustrates BER of the convolutional lattice in *Example 2*: The overall BER is shown by a black solid line marked with (x). The BER of the outer layer is nearly equal to the overall BER because the performance of the code is bounded by the MED of the outer layer (note that minimum Euclidean distance of the outer layer is smaller than minimum Euclidean distance of the inner layer). Clearly, due to the large minimum Euclidean distance of inner layer, its BER is much lower.

It was claimed earlier (without proof) that neglecting the MED criterion of the definition of Construction D will degrade the error performance of the overall system. In order to validate this, we have provided another simulation in Figure 8.14 where $\mathcal{C}_2$ is a $[10, 300, 12]$. The minimum Euclidean distance is 12 ($d_{\text{min}}^{\mathcal{C}_2} = 12$) which is smaller than the minimum Euclidean distance of the $\mathcal{C}_2$ in the above example. As expected, the BER of both layers is degraded when compared with Figure 8.13.

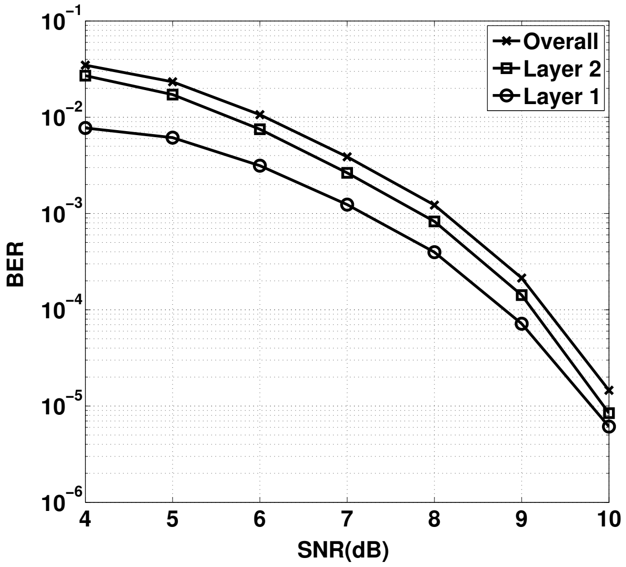


Figure 8.14: BER – Construction D (neglecting the MED criterion).

8.2.7 Conclusion and future work

Constructing convolutional lattices based on Construction A/D is proposed in this section. Also, lattice decoding using trellis structure of the underlying convolutional code is discussed. Unlike the existing lattice decoding algorithm the proposed method is practically feasible with reasonable complexity at arbitrarily high dimensions. Moreover, the performance of the proposed lattice decoder is found to be superior, since decoding is performed at lower dimension compared to the dimension of Construction A/D. Furthermore, the statistical characteristics of MLAN are derived in this section, and are exploited by the BCJR decoder.
Exploitation of the code lattices studied in this section and BCJR decoding algorithm on communication systems based on CaF is an ongoing research.

8.2.8 Distribution of N'

Assume T is a normally distributed random variable with mean equal to $\mu = \frac{a+b}{2}$ and variance equal to $\alpha^2\sigma^2$, i.e. $f_T(t) = \frac{1}{\sqrt{2\pi}\alpha\sigma}\mathrm{e}^{-\frac{(t-\mu)^2}{2\alpha^2\sigma^2}}$. Also, assume V is a random variable according to a uniform distribution in the (η_b, η_a) interval, i.e.

$$f_V(v) = \begin{cases} \frac{1}{\eta_a - \eta_b} & \eta_b < v < \eta_a \\ 0 & \text{otherwise} \end{cases} \tag{8.69}$$

and so, the pdf of $W = T + V$ is

$$f_W(w) \quad = \quad \frac{1}{2(\eta_a - \eta_b)}\mathrm{erf}\left(\frac{2w - b - a(3-2\alpha)}{2\sqrt{2}\alpha\sigma}, \frac{2w - a - b(3-2\alpha)}{2\sqrt{2}\alpha\sigma}\right)$$

with $\eta_i = i(1-\alpha)$.

Proof. Assuming $W = T + V$, the pdf of W is the convolution of the pdf of T and V, i.e. $f_W(w) = f_T(t) * f_V(v)$. By resorting to the definition of the convolution operator, one can write

$$\begin{aligned} f_W(w) &= \int_{-\infty}^{\infty} f_V(t) f_T(w-t)\mathrm{d}t \\ &= \frac{1}{2(\eta_a - \eta_b)} \int_{\eta_b}^{\eta_a} \frac{1}{\sqrt{2\pi}\alpha\sigma} \mathrm{e}^{-\frac{(w-\mu-t)^2}{2\alpha^2\sigma^2}} \mathrm{d}t. \end{aligned} \tag{8.70}$$

Considering that $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \mathrm{e}^{-t^2}\mathrm{d}t$ and $\mu = \frac{a+b}{2}$, after some manipulation, (8.70) can easily be simplified according to

$$f_W(w) \quad = \quad \frac{1}{2(\eta_a - \eta_b)}\mathrm{erf}\left(\frac{2w - b - a(3-2\alpha)}{2\sqrt{2}\alpha\sigma}, \frac{2w - a - b(3-2\alpha)}{2\sqrt{2}\alpha\sigma}\right). \tag{8.71}$$

□

9 Conclusions

This final version of the deliverable on “Terminal node processing for advanced scenarios” provides a comprehensive technical overview of selected physical layer techniques and algorithms which has been successfully implemented by the HW and SLS demonstrators (as reported in the WP5 deliverables). In addition, it introduces some additional promising physical layer techniques and algorithms developed in the last stage of the project.

Some particular algorithms/techniques presented in this report have already been successfully implemented by the HW/SLS demonstrator(s) as we summarise below:

Coded superposition modulation technique (sections 2.1, 2.2) was successfully implemented by the SMN HW demonstrator, as reported in [D5.42]. Appropriate modifications of the node processing strategies were implemented to combat the real-world setup impairments like the non-coherent complex phase rotation of individual source transmissions or direct channel availability between the source and its desired destination. The resulting algorithm is eligible to outperform conventional (i.e. non WPLNC) relaying strategies in a 5-node single relay network topology.

Cloud access node scheduling strategies (section 4.3) which interpret the cloud as a macro-relay using some approximations for parallel and mutually interacting flows have been investigated within the SLS demonstrating activity (see [23]). Even if the validations are far from being exhaustive, this activity gives the guidelines on new scheduling strategies to access to sub-networks as macro-relays.

Bibliography

[1] J. Sýkora and A. G. Burr, "Layered design of hierarchical exclusive codebook and its capacity regions for HDF strategy in parametric wireless 2-WRC," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3241–3252, Sep. 2011.

[2] ——, "Hierarchical alphabet and parametric channel constrained capacity regions for HDF strategy in parametric wireless 2-WRC," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2010.

[3] ——, "Network coded modulation with partial side-information and hierarchical decode and forward relay sharing in multi-source wireless network," in *Proceedings of the European Wireless Conference (EW)*, Apr. 2010.

[4] T. Uřičář, J. Sýkora, B. Qian, and W. H. Mow, "Superposition coding for wireless butterfly network with partial network side-information," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013.

[5] T. Uřičář and J. Sýkora, "Non-uniform 2-slot constellations for relaying in butterfly network with imperfect side information," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1369–1372, Sep. 2012.

[6] ——, "Design criteria for hierarchical exclusive code with parameter-invariant decision regions for wireless 2-way relay channel," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 921427, pp. 1–13, Jul. 2010.

[7] M. Hekrdla and J. Sýkora, "Design of uniformly most powerful alphabets for HDF 2-way relaying employing non-linear frequency modulations," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 128, pp. 1–18, Oct. 2011.

[8] T. Uřičář and J. Sýkora, "Non-uniform 2-slot constellations for bidirectional relaying in fading channels," *IEEE Communications Letters*, vol. 15, no. 8, pp. 795–797, Aug. 2011.

[9] T. Koike-Akino, P. Popovski, and V. Tarokh, "Denoising maps and constellations for wireless network coding in two-way relaying systems," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2008.

[10] ——, "Denoising strategy for convolutionally-coded bidirectional relaying," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Jun. 2009.

[11] ——, “Optimized constellations for two-way wireless relaying with physical network coding,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 773–787, Jun. 2009.

[12] P. Popovski and T. Koike-Akino, “Coded bidirectional relaying in wireless networks,” in *New Directions in Wireless Communications Research*, V. Tarokh, Ed. Springer, Sep. 2009, ch. 11, pp. 291–316.

[13] S. Zhang and S.-C. Liew, “Channel coding and decoding in a relay system operated with physical-layer network coding,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 788–796, Jun. 2009.

[14] U. Erez, S. Shamai (Shitz), and R. Zamir, “Capacity and lattice strategies for canceling known interference,” *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3820–3833, Nov. 2005.

[15] M. P. Wilson, K. Narayanan, H. D. Pfister, and A. Sprintson, “Joint physical layer coding and network coding for bidirectional relaying,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5641–5654, Nov. 2010.

[16] U. Erez and R. Zamir, “Achieving 1/2 log(1+SNR) on the AWGN channel with lattice encoding and decoding,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[17] R. Zamir, “Lattices are everywhere,” in *Proceedings of the Information Theory and Applications Workshop (ITA)*, Feb. 2009.

[18] N. Sommer, M. Feder, and O. Shalvi, “Low-density lattice codes,” *IEEE TTransactions on Information Theory*, vol. 54, no. 4, pp. 1561–1585, Apr. 2008.

[19] W. Nam, S.-Y. Chung, and Y. H. Lee, “Capacity bounds for two-way relay channels,” in *Proceedings of the International Zurich Seminar on Communications (IZS)*, Mar. 2008.

[20] B. Nazer and M. Gastpar, “Reliable physical layer network coding,” *Proceedings of the IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011.

[21] S.-C. Liew, S. Zhang, and L. Lu, “Physical-layer network coding: Tutorial, survey, and beyond,” *Elsevier Physical Communication*, vol. 6, pp. 4–42, Mar. 2013.

[22] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[23] A. Parichehreh, K. Ramantas, U. Spagnolini, and J. S. Vardakasy, “Scheduling of the super-dense wireless cloud networks,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2015.

[24] P. Popovski and H. Yomo, "Physical network coding in two-way wireless realy channels," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2007.

[25] J. Liu, M. Tao, and Y. Xu, "Pseudo exclusive-OR for LDPC coded two-way relay block fading channels," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Jun. 2011.

[26] K. Yasami, A. Razi, and A. Abedi, "Analysis of channel estimation error in physical layer network coding," *IEEE Communications Letters*, vol. 15, no. 10, pp. 1029–1031, Oct. 2011.

[27] S. Zhang, S.-C. Liew, and P. P. K. Lam, "On the synchronization of physical-layer network coding," in *Proceedings of the Information Theory Workshop (ITW)*, Mar. 2006.

[28] F. Rossetto and M. Zorzi, "On the design of practical asynchronous physical layer network coding," in *Proceedings of the IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Sep. 2009.

[29] Y. Huang, S. Wang, Q. Song, L. Guo, and A. Jamalipour, "Synchronous physical-layer network coding: A feasibility study," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 1536–1276, Aug. 2013.

[30] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference with structured codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Jul. 2008.

[31] T. Uřičář, J. Sýkora, B. Qian, and W. H. Mow, "Wireless (physical layer) network coding with limited hierarchical side-information: Maximal sum-rates in 5-node butterfly network," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5582–5595, Oct. 2014.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* John Wiley & Sons, 1991.

[33] T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 49–60, Jan. 1981.

[34] P. A. Höher and T. Wo, "Superposition modulation: Myth and facts," *IEEE Communications Magazine*, vol. 49, no. 12, pp. 110–116, Dec. 2011.

[35] M. Hekrdla and J. Sýkora, "On indexing of lattice-constellations for wireless network coding with modulo-sum decoding," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, 2013.

[36] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, Jul. 1999.

[37] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory," in *Foundations and Trends in Communications and Information Theory.* now Publishers, Jul. 2006, vol. 2, no. 5, pp. 330–381.

[38] E. Biglieri, *Coding for wireless channels.* Springer, 2010.

[39] J. Harshan and B. S. Rajan, "Finite signal-set capacity of two-user Gaussian multiple access channel," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Jul. 2008.

[40] ——, "On two-user Gaussian multiple access channels with finite input constellations," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1299–1327, Mar. 2011.

[41] S. Zhang and S.-C. Liew, "Applying physical-layer network coding in wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 870268, pp. 1–12, Mar. 2010.

[42] A. Li, Y. Yan, and H. Kayama, "An enhanced denoise-and-forward relaying scheme for fading channel with low computational complexity," *IEEE Signal Processing Letters*, vol. 15, no. 11, pp. 857–860, Nov. 2008.

[43] C. B. Schlegel and L. C. Perez, *Trellis and Turbo Coding.* John Wiley & Sons, 2004.

[44] R. F. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[45] F. H. P. Fitzek, J. Heide, M. V. Pedersen, and M. D. Katz, "Implementation of network coding for social mobile clouds [Applications corner]," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 159–164, Jan. 2013.

[46] A. Paramanathan, M. V. Pedersen, D. E. Lucani, F. H. P. Fitzek, and M. D. Katz, "Lean and mean: Network coding for commercial devices," *IEEE Wireless Communications*, vol. 20, no. 5, pp. 54–61, Oct. 2013.

[47] S. Zhang, S.-C. Liew, and P. P. K. Lam, "Hot topic: Physical-layer network coding," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, Sep. 2006.

[48] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011.

[49] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.

[50] Z. Chen, B. Xia, Z. Hu, and H. Liu, “Design and analysis of multi-level physical-layer network coding for Gaussian two-way relay channels,” *IEEE Transactions on Communications*, vol. 62, no. 6, pp. 1803–1817, Jun. 2014.

[51] J. Richter, C. Scheunert, and E. A. Jorswieck, “An efficient branch-and-bound algorithm for compute-and-forward,” in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) – Second Workshop on Network Coding in Wireless Relay Networks*, Jun. 2012.

[52] S. Sahraei and M. Gastpar, “Compute-and-forward: Finding the best equation,” Oct. 2014.

[53] L. Wei and W. Chen, “Compute-and-forward network coding design over multi-source multi-relay channels,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 9, pp. 3348–3357, Sep. 2012.

[54] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups.* Springer, 1999.

[55] U. Erez, S. Litsyn, and R. Zamir, “Lattices which are good for (almost) everything,” *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3401–3416, Oct. 2005.

[56] B. Nazer, “Successive compute-and-forward,” in *Proceedings of the International Zurich Seminar on Communication (IZS)*, Mar. 2012.

[57] J. Sýkora and J. Hejtmánek, “Joint and recursive hierarchical interference cancellation with successive compute & forward decoding,” in *Proceedings of the Management Committee Meeting, COST IC1004*, May 2014.

[58] G. D. Forney, M. D. Trott, and S.-Y. Chung, “Sphere-bound-achieving coset codes and multilevel coset codes,” *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 820–850, May 2000.

[59] Y. Wang and A. G. Burr, “Complex low density lattice codes to lattice network coding,” in *Proceedings of the IEEE International Communications Conference (ICC)*, Jun. 2015.

[60] Q. T. Sun, J. Yuan, T. Huang, and K. W. Shum, “Lattice network codes based on Eisenstein integers,” *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2713–2725, Jul. 2013.

[61] N. E. Tunali, K. R. Narayanan, J. J. Boutros, and Y.-C. Huang, “Lattices over Eisenstein integers for compute-and-forward,” in *Proceedings of the Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct. 2012.

[62] Y.-C. Huang and K. R. Narayanan, “Multistage compute-and-forward with multilevel lattice codes based on product constructions,” in *Proceedings of the IEEE Symposium on Information Theory (ISIT)*, Jun. 2014.

[63] D. G. Northcott, *Lessons on Rings, Modules and Multiplicities*, 1968.

[64] C. Feng, R. W. Nobrega, F. R. Kschischang, and D. Silva, "Communication over finite-chain-ring matrix channels," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5899–5917, Oct. 2014.

[65] B. R. McDonald, *Finite Rings with Identity.* Marcel Dekker Inc, 1974.

[66] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups.* Springer, 1998.

[67] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, Jul. 1999.

[68] A. G. Burr and T. J. Lunn, "Block-coded modulation optimized for finite error rate on the white Gaussian noise channel," *IEEE Transactions on Information Theory*, vol. 43, no. 1, pp. 373–385, Jan. 1997.

[69] Y. H. Gan, C. Ling, and W. H. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2701–2710, Jul. 2009.

[70] Y. Wang and A. G. Burr, "Code design for iterative decoding of multilevel codes," *IEEE Transactions on Communications*, vol. 63, no. 7, pp. 2404–2419, Jul. 2015.

[71] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[72] J. Kliewer, S. X. Ng, and L. Hanzo, "Efficient computation of EXIT functions for nonbinary iterative decoding," *IEEE Transactions on Communications*, vol. 54, no. 12, pp. 2133–2136, Dec. 2006.

[73] Y. Yona and M. Feder, "Complex low density lattice codes," in *Proceedings of the IEEE International Symposium on Information Theory Proceedings (ISIT)*, Jun. 2010.

[74] O. Shalvi, N. Sommer, and M. Feder, "Signal codes: Convolutional lattice codes," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5203–5226, Aug. 2011.

[75] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 773–787, Jun. 2009.

[76] S. Fu, K. Lu, T. Zhang, Y. Qian, and H.-H. Chen, "Cooperative wireless networks based on physical layer network coding," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 86–95, Dec. 2010.

[77] S. H. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.

[78] J. Sýkora and A. G. Burr, “Layered design of hierarchical exclusive codebook and its capacity regions for HDF strategy in parametric wireless 2-WRC,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3241–3252, Sep. 2011.

[79] ——, “Advances in wireless network coding – the future of cloud communications,” in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2013.

[80] M. E. J. Newman, *Networks – An Introduction.* Oxford University Press, 2010.

[81] H. H. K. Lentz, T. Selhorst, and I. M. Sokolov, “Unfolding accessibility provides a macroscopic approach to temporal networks,” *Physical Review Letters, 118701*, vol. 110, no. 11, Mar. 2013.

[82] M. Xiao, J. Kliewer, and M. Skoglund, “Design of network codes for multiple-user multiple-relay wireless networks,” *IEEE Transactions on Communications*, vol. 60, no. 12, pp. 3755–3766, Dec. 2012.

[83] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[84] S. Savazzi, U. Spagnolini, L. Goratti, and S. Galimberti, “Wireless cloud networks for critical industrial quality control,” in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2013.

[85] T. Uřičář, T. Hynek, P. Procházka, and J. Sýkora, “Wireless-aware network coding: Solving a puzzle in acyclic multi-stage cloud networks,” in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2013.

[86] O. Georgiou, M. Z. Bocus, M. R. Rahman, C. P. Dettmann, and J. P. Coon, “Network connectivity in non-convex domains with reflections,” *IEEE Communications Letters*, vol. 19, no. 3, pp. 427–430, Mar. 2015.

[87] A. Parichehreh and U. Spagnolini, “Inter- and intra-cloud resource allocation for delay sensitive industrial networks,” in *Proceedings of the European Conference on Networks and Communications (EuCNC)*, Jun. 2014.

[88] Z. K. M. Ho, E. A. Jorswieck, and S. Engelmann, “Information leakage neutralization for the multi-antenna non-regenerative relay-assisted multi-carrier interference channel,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1672–1686, Sep. 2013.

[89] Z. Ding, I. Krikidis, B. Rong, J. S. Thompson, C. Wang, and S. Yang, “On combating the half-duplex constraint in modern cooperative networks: Protocols and techniques,” *Wireless Communications, IEEE*, vol. 19, no. 6, pp. 20–27, Dec. 2012.

[90] V. Angelakis, A. Ephremides, Q. He, and D. Yuan, "Minimum-time link scheduling for emptying wireless systems: Solution characterization and algorithmic framework," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1083–1100, Feb. 2014.

[91] B. Barua, M. Abolhasan, D. R. Franklin, and F. Safaei, "Outage probability of multihop relay networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jul. 2013.

[92] R. Babaee and N. C. Beaulieu, "Cross-layer design for multihop wireless relaying networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3522–3531, Nov. 2010.

[93] E. C. van der Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, Spring 1971.

[94] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.

[95] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity – Part I. System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[96] ——, "User cooperation diversity – Part II. Implementation aspects and performance analysis," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1939–1948, Nov. 2003.

[97] T. Cover and A. El Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, Sep. 1979.

[98] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, Nov. 1999.

[99] W. Guan and H. Luo, "Joint MMSE transceiver design in non-regenerative MIMO relay systems," *IEEE Communications Letters*, vol. 12, no. 7, pp. 517–519, Jul. 2008.

[100] R. Mo and Y. Chew, "MMSE-based joint source and relay precoding design for amplify-and-forward MIMO relay networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4668–4676, Sep. 2009.

[101] M. R. A. Khandaker and Y. Rong, "Joint source and relay optimization for multiuser MIMO relay communication systems," in *Proceedings of the International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec. 2010.

[102] K. Cumanan, Y. Rahulamathavan, S. Lambotharan, and Z. Ding, "MMSE-based beamforming techniques for relay broadcast channels," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 4045–4051, Oct. 2013.

[103] M. Molu Mortazawi and N. Görtz, "Optimal precoding in the relay and the optimality of largest eigenmode relaying with statistical channel state information," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2113–2123, Apr. 2014.

[104] O. Munoz-Medina, J. Vidal, and A. Agustin, "Linear transceiver design in nonregenerative relays with channel state information," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2593–2604, Jun. 2007.

[105] X. Tang and Y. Hua, "Optimal design of non-regenerative MIMO wireless relays," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1398–1407, Apr. 2007.

[106] P. Dharmawansa, M. R. McKay, R. K. Mallik, and K. Ben Letaief, "Ergodic capacity and beamforming optimality for multi-antenna relaying with statistical CSI," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2119–2131, Aug. 2011.

[107] C. Jeong, B. Seo, S. R. Lee, H.-M. Kim, and I.-M. Kim, "Relay precoding for non-regenerative MIMO relay systems with partial CSI feedback," *IEEE Transactions on Wireless Communications*, vol. 11, no. 5, pp. 1698–1711, May 2012.

[108] G. Farhadi and N. C. Beaulieu, "On the performance of Amplify-and-Forward cooperative systems with fixed gain relays," *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1851–1856, May 2008.

[109] D. S. Shiu, *Wireless Communication Using Dual Antenna Arrays.* Kluwer Academic Publishers, 1999.

[110] D.-S. Shiu, G. J. Foschini, M. J. Gans, and J. M. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 502–513, Mar. 2000.

[111] A. Zappone, P. Cao, and E. A. Jorswieck, "Energy efficiency optimization in relay-assisted MIMO systems with perfect and statistical CSI," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 443–457, Jan. 2014.

[112] M. O. Hasna and M.-S. Alouini, "A performance study of dual-hop transmissions with fixed gain relays," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2003.

[113] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products.* Academic, 2007.

[114] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes.* McGraw Hill, 2002.

[115] A. G. Burr and J. Sýkora, "Extended mappings for wireless network coded butterfly network," in *Proceedings of the European Wireless Conference (EW)*, Apr. 2011.

[116] T. Uřičář and J. Sýkora, "Systematic design of hierarchical network code mapper for butterfly network relaying," in *Proceedings of the European Wireless Conference (EW)*, Apr. 2012.

[117] O. Ordentlich, U. Erez, and B. Nazer, "The approximate sum capacity of the symmetric Gaussian K-user interference channel," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3450–3482, Jun. 2014.

[118] I.-J. Baik and S.-Y. Chung, "Network coding for two-way relay channels using lattices," in *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2008.

[119] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall, 1993.

[120] A. Y.-C. Peng, S. Yousefi, and I.-M. Kim, "On error analysis and distributed phase steering for wireless network coding over fading channels," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, pp. 5639–5649, Nov. 2009.

[121] P. Popovski and H. Yomo, "Bi-directional amplification of throughput in a wireless multi-hop network," in *Proceedings of the IEEE Vehicular Technology Conference (VTC 2006-Spring)*, May 2006.

[122] Q. T. Sun, T. Huang, and J. Yuan, "On lattice-partition-based physical-layer network coding over GF(4)," *IEEE Communications Letters*, vol. 17, no. 10, pp. 1988–1991, Oct. 2013.

[123] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," in *Proceedings of the IEEE International Symposium on Information Theory Proceedings (ISIT)*, Jun. 2010.

[124] Y. Wang and A. G. Burr, "Physical-layer network coding via low density lattice codes," in *Proceedings of the European Conference on Networks and Communications (EuCNC)*, Jun. 2014.

[125] Y. Yona and M. Feder, "Efficient parametric decoder of low density lattice codes," in *Proceedings of the IEEE International Symposium on Information Theory Proceedings (ISIT)*, Jun. 2009.

[126] B. Kurkoski and J. Dauwels, "Message-passing decoding of lattices using Gaussian mixtures," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Jul. 2008.

[127] N. Sommer, M. Feder, and O. Shalvi, "Shaping methods for low-density lattice codes," in *Proceedings of the IEEE Information Theory Workshop (ITW)*, Oct. 2009.

[128] A. Sakzad, M. R. Sadeghi, and D. Panario, "Turbo lattices: Construction and error decoding performance," *CoRR*, vol. abs/1108.1873, 2011. [Online]. Available: http://arxiv.org/abs/1108.1873

[129] A. Sakzad, M.-R. Sadeghi, and D. Panario, "Construction of turbo lattices," in *Proceedings of the Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2010.

[130] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[131] A. Ghasemmehdi and E. Agrell, "Faster recursions in sphere decoding," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3530–3536, 2011.

[132] E. Viterbo and E. Biglieri, "Computing the Voronoi cell of a lattice: The diamond-cutting algorithm," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 161–171, Jan. 1996.

[133] M. Molu Mortazawi, "Reduced complexity BCJR algorithms for Faster than Nyquist (FTN) signalling," *MSc Thesis, Lund University*, Jun. 2009.

[134] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 409–417, Mar. 1994.

[135] G. W. W. Charles H. Swannack, Uri Erez, "Geometric relationships between Gaussian and modulo-lattice error exponents," *CoRR*, vol. abs/1308.1609, 2013. [Online]. Available: http://arxiv.org/abs/1308.1609

[136] T. Liu, P. Moulin, and R. Kötter, "On error exponents of modulo lattice additive noise channels," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 454–471, Feb. 2006.

[137] M. Molu Mortazawi and N. Görtz, "Optimal precoding in the relay and the optimality of largest eigenmode relaying with statistical channel state information," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2113–2123, Apr. 2014.

[138] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm – I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[139] H. Vikalo and B. Hassibi, "On the sphere-decoding algorithm – II. Generalizations, second-order statistics, and applications to communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2819–2834, Aug. 2005.

[140] J. Forney, G. D., "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Transactions on Information Theory*, vol. 35, no. 4, pp. 906–909, Jul. 1989.

[141] M.-R. Sadeghi, A. H. Banihashemi, and D. Panario, "Low-density parity-check lattices: Construction and decoding analysis," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4481–4495, Oct. 2006.

[142] R. H. Morelos-Zaragoza and H. Imai, "Binary multilevel convolutional codes with unequal error protection capabilities," *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 850–853, Jul. 1998.

[143] J. Harshan, E. Viterbo, and J.-C. Belfiore, "Construction of Barnes-Wall lattices from linear codes over rings," in *Proceedings of the IEEE International Symposium on Information Theory Proceedings (ISIT)*, Jul. 2012.

[144] ——, "Practical encoders and decoders for Euclidean codes from Barnes-Wall lattices," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4417–4427, Nov. 2013.

[145] Y. Yan, C. Ling, and X. Wu, "Polar lattices: Where Arikan meets Forney," in *Proceedings of the IEEE International Symposium on Information Theory Proceedings (ISIT)*, Jul. 2013.

[146] W. Kositwattanarerk and F. Oggier, "Connections between Construction D and related constructions of lattices," *CoRR*, vol. abs/1308.6175, 2013. [Online]. Available: http://arxiv.org/abs/1308.6175