Université de Cergy-Pontoise
École doctorale Sciences et Ingénierie

THÈSE

pour obtenir le titre de

Docteur en Sciences

Spécialité Sciences et Technologies
de l'Information et de la Communication

par

Anne Savard

Coding for cooperative communications: Topics in distributed source coding and relay channels

Thèse dirigée par David Declercq
Co-dirigée par Claudio Weidmann

préparée à ETIS dans l'équipe ICI

Soutenue le 22 Septembre 2015

Devant la Commission d'Examen

Jury

Rapporteur	: Matthieu Bloch	- Georgia Tech, Atlanta
Rapporteur	: Michael Gastpar	- EPFL
Président de jury	: Michel Kieffer	- LSS
Examinateur	: Ingmar Land	- Huawei France
Directeur	: David Declercq	- ETIS
Co-directeur	: Claudio Weidmann	- ETIS

Remerciements

Je tiens tout d'abord à remercier Claudio Weidmann et David Declercq pour avoir encadré cette thèse. Leur disponibilité, leur aide, leur soutien et la confiance qu'ils ont bien voulu m'accorder a permis de réaliser et d'enrichir les travaux de ce manuscrit. De même, je souhaite remercier Inbar Fijalkow et Mathias Quoy pour m'avoir accueillie pendant trois ans au sein du laboratoire ETIS et Annick Bertinotti, Sokhena Men ainsi que Nelly Bonard pour leur aide précieuse dans toutes les démarches administratives.

Je remercie également Matthieu Bloch et Michael Gastpar pour avoir été rapporteur. Leurs remarques et commentaires m'ont aidé à améliorer la qualité de ce document. Je tiens aussi à remercier Ingmar Land pour avoir accepté de faire partie de mon jury et Michel Kieffer pour l'avoir présidé.

Une thèse, c'est bien évidemment trois ans de travail mais pas que.

Je tiens à vivement remercier les personnes avec qui j'ai passé la quasi-totalité des pauses midi, à savoir Veronica, Romain N., Olivier, David P., Laura, Laurent, Elsa et Maël. Grâce à vous, j'ai tous les jours pu faire une pause bien animée dans la journée.

Une thèse, c'est aussi passer du temps avec les autres doctorants, tant au laboratoire qu'en dehors. Un grand merci à tous, et plus particulièrement à Alexis L., Romain T., Liang, Voisin et Laurent pour leur aide et leur soutien. J'en profite pour remercier Voisin et Javier, mes co-bureau, pour leur amitié et les pauses thé.

Une thèse c'est aussi acquérir des compétences dans différents domaines, acquérir une ouverture d'esprit. Je tiens tout particulièrement à remercier Veronica et Maël, pour toute l'aide que vous avez pu m'apporter tant en maths que pour l'esthétique et la répétition des présentations.

De même, un grand merci à tous ceux qui m'ont aidé à préparer les cours que j'ai eu la chance de donner à l'ENSEA. Merci pour votre confiance et vos conseils précieux.

Un grand merci aux étudiants d'ARES, passer du temps avec vous m'a beaucoup aidé. Sur le même plan, un grand merci à Voisin et Laurent avec qui nous avons fondé l'association NOVA-Robotics afin d'oublier le temps des week-ends nos thèses respectives pour nous concentrer sur la robotique et à Antoine et Veronica pour nous avoir rejoint.

Bien sûr, ces trois années ne se limitent pas au temps passé au laboratoire. Un grand merci à ma famille et à mes amis pour avoir cru en moi. Sans vous et votre soutien sans faille, cette thèse n'aurait peut-être pas vu le jour. Plus particulièrement, merci à Vincent, Laurent, Veronica, Voisin, Alexis A., Romain C., Liang et Flo pour les sorties ciné, musées, restos, expos, les soirées au téléphone, les virées shopping, la reprise du dessin et tout le temps passé à expérimenter de la pâtisserie.

Le dernier des remerciements, et non des moindres, va à mon meilleur ami. Merci d'avoir toujours répondu présent, d'avoir toujours cru en moi et de m'avoir toujours soutenue, dans les moments de doute mais aussi de joie.

Contents

List of Figures

List of Tables

Acronyms and notations

AF	Amplify-and-Forward
APP	A Posteriori Probabilities
AWGN	Additive White Gaussian Noise
BP	Belief-Propagation
BSC	Binary Symmetric Channel
CDF	Compress/Decode-and-Forward
CF	Compress-and-Forward
CN	Check Node
CoF	Compute-and-Forward
CSB	Cut-Set Bound
DF	Decode-and-Forward
FSM	Finite State Machine
LDPC	Low-Density Parity Check
LLR	Log-Likelihood Ratio
MAC	Multiple Access Channel
MMSE	Minimum Mean-Squared Error
mRC	Multiway Relay Channel
pdf	Probability distribution function
PEG	Progressive Edge Growth
RC	Relay Channel
SI	Side Information
TWRC	Two Way-Relay Channel
VN	Variable Node
$\|\mathcal{A}\|$	Cardinality of the set $\mathcal{A}$
$\mathcal{A} \setminus b$	Subset $\mathcal{A}$ without element b
$\overline{x} = 1 - x$	
$a \propto b$	Proportional operator

$\bigotimes$	Convolution operator
$\oplus$	Component-wise modulo-2 adder
$\log_2^+(x) = \max(0, \log_2(x))$	
$C(x) = \frac{1}{2}\log_2(1+x)$	Gaussian capacity function
d_H	Hamming distance
$\mathbb{E}$	Expectation
$GF(2)$	Galois Field of order 2
H	Entropy function
H_2	Binary Entropy function: $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$
I	Mutual information
Var	Variance
$X - Y - U$	Markov chain

Chapter 0

Foreword

0.1 Motivation

In the last decades, our relation to telecommunications has changed. The goal is now to send a growing amount of data over wireless networks. Moreover, while twenty years ago the number of wireless communicating devices was very low, now almost everyone has such a device. Due to the nature of the wireless medium, devices can overhear signals sent by neighbouring devices. Thus, since users wish to send more and more data, with a reduced power, one widely accepted solution is to consider cooperative communications.

In this thesis, we investigate some multi-terminal information theory problems. The simplest building block for cooperative communications is the relay channel. In this model, represented on Figure 1, a user wishes to send its data to a destination with the help of a relay.

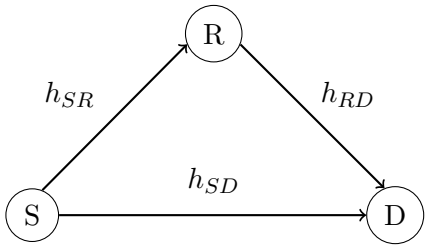


Figure 1: Relay channel

We first focus on two somewhat dual problems: one in source coding and one in channel coding. On one hand, we consider the binary source coding problem with coded side information, where nodes have correlated inputs, and on the other hand, the Gaussian relay channel with correlated noises.

Moreover, the source coding with coded side information problem is equivalent to the information bottleneck problem studied in [Tishby et al., 1999], which is a typical problem of cooperative communication, since the helper (that can be seen as a relay) tries to extract from its observation, as much information on the source message as possible, while keeping its rate constraint satisfied.

For the source coding part, convolutional codes and Low-Density Parity Check (LDPC) codes are used. Since it is more easy to optimize LDPC codes, we choose a specific convolutional code and optimize the LDPC code accordingly.

For the channel coding part, we use either lattice coding or Additive White Gaussian Noise (AWGN) coding.

In the last part of the thesis, we focus on a more realistic multi-user communication scenario, where N users, grouped into L clusters of K users each wish to exchange their messages within a cluster with the help of a single relay. This model, called multiway relay channel, has been proposed by [Gündüz et al., Jan. 2013]. Here we propose an extension of this model by adding direct links between the users of a cluster, which seems to be a more realistic model of users

overhearing each other. The aim of this part is not to propose new relaying schemes per se, but to show that standard schemes based either on lattice coding or AWGN coding can be used also on this extended multiway relay channel model.

0.2 Thesis outline and contributions

This thesis is organized into two parts: Part I presents source coding with coded side information and Part II focuses on relay channels.

0.2.1 Part I

The first part is dedicated to the source coding problem with coded side information. We propose a practical solution using a convolutional quantizer for the coded side information branch and a Low-Density Parity Check (LDPC) code on the main branch. Since many sequences are mapped onto the same quantization index, instead of using the reconstruction of the quantization index as a single representative side information, one can modify it by projecting an intermediate estimate of the source word onto the Voronoi cell associated to this index. Our main contributions to this consist in characterizing the Voronoi cell $\mathcal{V}_0$ of a convolutional quantizer using a Finite-State Machine (FSM) and then proposing a new decoding principle based on this characterization, improving thus the performance obtained for the coded side information problem.

Chapter 1 presents the theoretical aspect of source coding with coded side information for the binary case and a state-of-the-art LDPC code based algorithm used for this problem [Liveris et al., 2002], as well as LDPC codes and convolutional quantizers.

Chapter 2 presents, in a tutorial way, how to characterize the Voronoi cell $\mathcal{V}_0$ of a convolutional code using a FSM and how to assign transition probabilities on this FSM.

Chapter 3 presents our improved decoder, based on a geometrical intuition. We present two main implementations of our method and finally show how to optimize the LDPC code used.

Publications The work presented in this part was presented in

- [Savard and Weidmann, 2013a] **A. Savard** and C. Weidmann, **"Décodeur amélioré pour le codage de source avec information adjacente compressée"** *GRETSI, Brest, France, 2013*
- [Savard and Weidmann, 2013b] **A. Savard** and C. Weidmann, **"Improved decoding for binary source coding with coded side information"** *Proc. IEEE Information Theory Workshop (ITW), Seville, Spain, 2013*
- [Savard and Weidmann, 2014a] **A. Savard** and C. Weidmann, **"Optimized codes for the binary coded side information problem"** *International Symposium on Turbo Codes (ISTC), Bremen, Germany, 2014*
- [Savard and Weidmann, 2015c] **A. Savard** and C. Weidmann, **"On the Hamming-space Voronoi regions of convolutional codes with applications"** *submitted to IEEE Transactions on Communication*

0.2.2 Part II

Part II of this thesis focuses on various relay channel models. We study extensions of the relay channel models, for which no achievable schemes were proposed so far. The goal is to characterize achievable rate regions using different state-of-the-art relaying schemes, as for instance Decode-and-Forward (DF), Compress-and-Forward (CF) or Amplify-and-Forward (AF). In order to compare these protocols, we derive cut-set bounds for each relay channel model, in order to obtain an upper bound on the capacity.

Chapter II introduces the different relay channel models considered in this thesis.

Chapter 4 presents two different Gaussian Relay Channel models: the first one is the standard full-duplex model and the second one is a more general model considering correlated noises at the relay and the destination.

We briefly present achievable rate regions for the first model that can be found in the literature. Furthermore, we introduce standard relaying schemes such as DF, AF and CF. Proofs, even if standard, are presented in this thesis since we will refer to them when proving achievability for more elaborated relay models.

We then show that lattices can achieve a previously known achievable rate region of the Gaussian relay channel with correlated noises.

Chapter 5 presents a first extension of the Gaussian relay channel by considering Two-way relay channel, a multi-hop communication scenario, in which two users wish to exchange data with the help of a relay. We study three different Gaussian two-way relay channel models: the standard full-duplex two-way relay channel without and with direct links, and a more general full-duplex two-way relay channel with direct links and correlated noises at the relay and destinations. In this chapter, we derive a full-duplex CF lattice based scheme for the two-way relay channel without direct links and then we propose a Compress/Decode-and-Forward scheme for the two-way relay channel with direct links and correlated noises, based on the CF scheme proposed in previous chapter for the relay channel with correlated noises.

Chapter 6 presents a more general extension of the Gaussian relay channel, the multi-way relay channel, by considering multiple clusters of users that wish to exchange messages locally within each cluster, with the help of one single relay. This model is an extension of the one proposed by [Gündüz et al., Jan. 2013] by adding direct links between users of the same cluster. We first characterize achievable rates for various standard relaying protocols, when time-sharing among the cluster is used, and compare them with the rates obtained on Gündüz et al.'s model, which is considered as the asymptotic limit of our model when the gain on the relay-user links grows large. We also characterize the gap to the cut-set bound for the proposed protocols. The last part of this chapter gives results for the multi-way relay channel without the time-sharing assumption.

Publications The work presented in this part was presented in

- [Savard and Weidmann, 2014b] **A. Savard** and C. Weidmann, **"On the multiway relay channel with direct links"** *Proc. IEEE Information Theory Workshop (ITW), Hobart, Tasmania (Australia), 2014*

- [Savard and Weidmann, 2015a] **A. Savard** and C. Weidmann, **"Canal à relais multi-directionnel avec liens directs"** *GRETSI, Lyon, France, 2015*
- [Savard and Weidmann, 2015b] **A. Savard** and C. Weidmann, **"Lattice coding for the Gaussian one- and two-way relay channels with correlated noises"** *Proc. IEEE International Symposium on Information Theory (ISIT), Hong-Kong, China, 2015*
- [Savard and Weidmann, 2015d] **A. Savard** and C. Weidmann, **"On the multiway relay channel with intra-cluster links"** *to be submitted to IEEE Transactions on Wireless Communication*

Part I

Source coding with coded side information

Chapter 1

Theory and practice of the binary coded side information problem

Contents

1.1 Theory of the binary case

In various situations, such as sensor networks, distributed coding or networks with relay, side information (SI) is available at the decoder. Usually this side information, that is correlated to the source, is compressed in a many-to-one fashion. In the first part of this thesis, we consider the *binary source coding with coded side information problem*, which is a typical example of such a situation as well as a cooperative communication problem.

Two discrete sources X and Y, with finite alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and joint distribution $P_{X,Y}$, are separately encoded by encoders E_X and E_Y at rates R_X and R_Y, respectively. Decoder D_X tries to reconstruct X losslessly as $\hat{X}$, while Y serves only as SI for decoding X and won't be reconstructed. This situation is depicted in Figure 1.1.

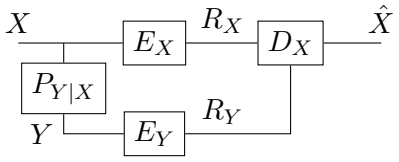


Figure 1.1: Coded side information problem: general case

The achievable rate region for this problem has been characterized by [Ahlswede and Körner, Nov. 1975]. A rate pair (R_X, R_Y) is achievable if

$$\begin{cases} R_X \geq H(X|U) \\ R_Y \geq I(Y;U), \end{cases} \tag{1.1}$$

for an auxiliary random variable $U \in \mathcal{U}$, with $|\mathcal{U}| \leq |\mathcal{Y}| + 2$, such that $X - Y - U$ form a Markov chain. H denotes the entropy function and I the mutual information.

In this thesis, we focus only on the binary case. For the case when both X and Y are binary symmetric, by [Gu et al., 2007] showed that encoding Y using binary quantization with Hamming distortion criterion is sufficient to achieve the rate region. In this case, one can give a closed-form expression of the achievable rate region. Let X be a Bernoulli-1/2 source. The correlation between X and Y is modeled by a Binary Symmetric Channel (BSC) with error probability p (BSC-p). Encoder E_Y produces a quantized version of Y and can be represented, in an idealized setting, by a BSC-D, where D is the optimal distortion obtained for rate R_Y. This is depicted in Figure 1.2.

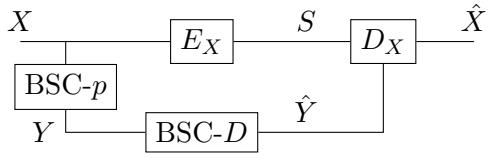


Figure 1.2: Coded side information problem: binary case

The achievable rate region (1.1) becomes :

$$\begin{cases} R_X \geq H_2(p + D - 2pD) \\ R_Y \geq 1 - H_2(D), \end{cases} \tag{1.2}$$

where H_2 denotes the binary entropy function.

If we compare this problem to the relay channel, we can note some similarities: Y, which is correlated with X serves only as side information, and won't be reconstructed as for instance the message sent by the relay in Compress-and-Forward, that only helps the decoding of the source message. The main difference with relay channel is that the messages sent from the source to the destination and to the 'helper node' (encoder E_Y) are not the same.

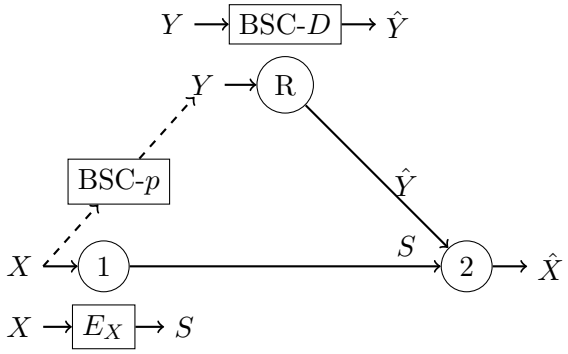


Figure 1.3: Coded side information problem from a relay channel point of view

If we take a closer look at the destination, we can note that from the main branch, it receives a compressed version S of X and from the SI branch a noisy version of X, thus this problem is equivalent to a Slepian-Wolf problem [Slepian and Wolf, 1973], for which a practical solution using Low-Density Parity Check (LDPC) codes has been proposed by [Liveris et al., 2002].

1.2 Overview of a practical scheme and outlook

As mentioned previously, since the concatenation of the two BSC is equivalent to a new BSC of error probability $\epsilon = p + D - 2pD$, our problem is equivalent to a Slepian-Wolf coding problem.

We present the following simple scheme based on a LDPC approach and a trellis quantizer for binary source coding with coded side information. This scheme will be referred throughout this thesis as standard method and will be used to compare the results obtained with the improved decoder and the optimized codes.

In the following, the sequence $\underline{x}$ will be modeled as coming from a Bernoulli-1/2 source. The correlation between $\underline{x}$ and the side information $\underline{y}$ will be modeled by a BSC-p.

The binary input sequence $\underline{x} = (x_1, x_2, \ldots, x_n)$ of length n is encoded by computing its syndrome $\underline{s} = \underline{x}H^T$, where H is the parity check matrix of an LDPC code. $\underline{y}$ is encoded to a quantized version w using a binary trellis-coded quantizer based on a convolutional code. The codeword $\hat{\underline{y}}$ closest in Hamming metric to the input sequence $\underline{y}$ is found using the Viterbi algorithm and the corresponding information sequence is output as index w. The reconstruction $\hat{\underline{y}}(w)$ associated to the index w is used as channel information at the LDPC decoder D_X (see Figure 1.2). This decoder must then estimate the sequence $\underline{x}$ from the syndrome $\underline{s}$ and the reconstructed SI $\hat{\underline{y}}(w)$ using the Belief Propagation (BP) algorithm as in [Liveris et al., 2002].

The key observation for the improved decoder presented in the first part of this thesis is the following: the coded side information is conveyed by a single quantization index w but many sequences are mapped onto this quantization index: they form the Voronoi cell $\mathcal{V}_w = \{y \in \{0,1\}^n | E_Y(y) = w\}$. In the practical LDPC-based solution, the iterative decoder only uses the codeword $\hat{Y}(w)$ associated to the quantization index w but choosing another sequence could achieve better performance. Our idea is to project an intermediate solution onto $\mathcal{V}_w$ and continue the decoding with this new sequence. The hope is that this new sequence is closer to X than $\hat{Y}(w)$ was, and could thus accelerate the decoding of X.

In order to perform this projection, we need a characterization of all Voronoi cells $\mathcal{V}_w$ associated to the used convolutional code. Since convolutional codes are linear, the goal is to write the set of all possible sequences $\mathcal{E}$ as the direct sum of the Voronoi cells $\mathcal{V}_w$ associated to each codeword w. The cells are such that $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$ and

$$\mathcal{E} = \bigcup_{c \in \mathcal{C}} \mathcal{V}_c = \bigcup_{c \in \mathcal{C}} \mathcal{V}_0 \oplus c,$$

where $\mathcal{C}$ is the code, $\oplus$ denotes the componentwise modulo-2 addition and $\mathcal{V}_0$ is the fundamental Hamming space Voronoi cell, i.e. the set of all sequences that are closer to the all-zero codeword than to any other codeword.

In the remaining of this chapter, we first introduce LDPC codes, as well as their usually associated iterative decoding algorithm (the Belief Propagation algorithm), and convolutional quantizers, as well as two algorithms, the Viterbi and the BCJR algorithms, usually used for their decoding.

1.3 Low-Density Parity Check code

1.3.1 Definition and decoding

In block coding, an information sequence is segmented into message blocks each of the length k. At the encoder, each input message $\underline{u} = [u_1, \ldots, u_k]$ is encoded into a longer binary sequence $\underline{v} = [v_1, \ldots, v_n]$, with $n > k$. The sequence $\underline{v}$ is called the *codeword* associated to the message u. The $n - k$ added bits are called the *redundancy*, they don't carry any new information but provide the capacity of detecting and/or correcting errors. The ratio k/n, that is called the *code rate*, represents the average number of information bits carried by each code bit.

Definition 1.3.1. *A binary block code of length n with 2^k codewords is called a (n,k) linear block code if and only if the 2^k codewords form a k-dimensional subspace of the vector space 2^n.*

Since a (n,k) linear block code $\mathcal{C}$ is a k-dimensional subspace of 2^n, there exist k linearly independent codewords $\underline{g}_1, \ldots, \underline{g}_k \in \mathcal{C}$ such that each codeword $v \in \mathcal{C}$ is a linear combination of these $\underline{g}_1, \ldots, \underline{g}_k$, which form a basis of $\mathcal{C}$. Using a matrix based notation, we may write G as $G = \begin{bmatrix} \underline{g}_1 \\ \vdots \\ \underline{g}_k \end{bmatrix}$. Thus, the codeword $\underline{v} = [v_1, \ldots, v_n]$ of the message $\underline{u} = [u_1, \ldots, u_k]$ can be expressed as $\underline{v} = \underline{u}G$.

Definition 1.3.2. *The above given matrix G is called a generator matrix of the (n,k) linear code $\mathcal{C}$.*

$$\mathcal{C} = \{\underline{v} \in 2^n | \forall \underline{u} \in 2^k, \underline{v} = \underline{u}G\}$$

Since a (n,k) linear block code $\mathcal{C}$ is a k-dimensional subspace of 2^n, its dual space is a $n-k$-dimensional subspace of 2^n, defined as $\mathcal{C}_d = \{\underline{w} \in 2^n | \forall \underline{v} \in \mathcal{C} < \underline{w}, \underline{v} >= 0\}$, where $< a, b >$ represents the inner product of a and b. $\mathcal{C}_d$ can thus be seen as a binary $(n, n-k)$ linear block code. As previously done, we can find a basis formed by $n-k$ linearly independent codewords $\underline{h}_1, \ldots, \underline{h}_{n-k}$ and define the generator matrix H of $\mathcal{C}_d$ as $H = \begin{bmatrix} \underline{h}_1 \\ \vdots \\ \underline{h}_{n-k} \end{bmatrix}$.

Based on these definitions, we can note that $HG^T = 0$.

Definition 1.3.3. *Let $\mathcal{C}$ be a (n,k) linear code and let $x \in 2^n$ be any vector. The subset*

$$x + \mathcal{C} = \{x + y | y \in \mathcal{C}\}$$

is called a coset of $\mathcal{C}$.

Definition 1.3.4. *A coset leader of coset $x + \mathcal{C}$ is a word $y \in x + \mathcal{C}$ of minimum weight.*

Definition 1.3.5. *The above given matrix H is called a parity check matrix of the (n,k) linear code $\mathcal{C}$.*

$$\mathcal{C} = \{\underline{v} \in 2^n | \underline{v}H^T = 0\}$$

In other words, a (n,k) linear block code $\mathcal{C}$ is defined as $\mathcal{C} = Im(G) = Ker(H)$.

LDPC codes have been introduced by Gallager in 1963 and rediscovered by MacKay in the 90s. They are linear block codes whose parity check matrix H is sparse, i.e. has a low density of 1s. The sparseness of H guarantees a decoding complexity which increases linearly with the code length. LDPC codes are very popular and widely used since their decoder performs near-capacity.

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

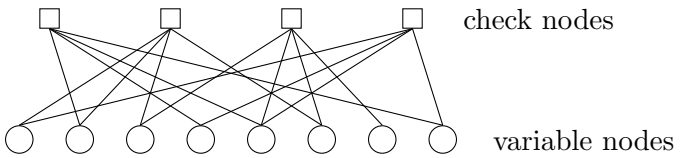


Figure 1.4: Example of a LDPC parity check matrix H and its associated Tanner graph

Definition 1.3.6. *Regular LDPC codes are such that each column of the parity check matrix is of weight w_c and each line of weight w_l. LDPC codes whose parity check matrix doesn't satisfy these criteria are irregular LDPC codes.*

LDPC codes can also be described by their Tanner graph. A Tanner graph is a bipartite graph, that is a graph composed of two different types of nodes: the variable nodes (VN) and the check nodes (CN). The n coordinates of the codewords are associated with the VN and each CN is associated with a parity check constraint, thus there are n VN and $n-k$ CN. A connection between CN i and VN j exists only if the entry h_{ij} of the matrix H is equal to 1, thus the adjacency matrix of this graph is the parity check matrix H of the code.

The degree of a node is the number of edges connected to it. For an irregular LDPC code, one defines two polynomials to describe the distribution of the edges as follows:

$$\lambda(x) = \sum_{i=1}^{V_{max}} \lambda_i x^{i-1} \text{ and } \rho(x) = \sum_{i=1}^{C_{max}} \rho_i x^{i-1},$$

where V_{max} and C_{max} are the maximal VN and CN degrees, respectively. Given the Tanner graph of the LDPC code, λ_i is the fraction of edges that are connected to a VN of degree i (i.e. connected to i CN). Similarly, ρ_i is the fraction of edges that are connected to a CN of degree i. The design rate of this code is given by:

$$R = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}.$$

The Tanner graph is also very useful for the decoding procedure: each node can be seen as an operating processor whose result is sent to its neighbors defined by the edges.

As said previously, LDPC codes can be decoded iteratively using their Tanner graph and message passing algorithms: Messages are passed along the edges of the graph. Messages from VN to CN are computed based on the observed values of the VN and some messages passed from neighboring CN. An important aspect of this algorithm is that the message sent by VN v to CN c must not take into account the incoming message from c to v. In the same way, messages from CN to VN are computed based on some messages sent by the neighboring VN.

One important algorithm is the Belief Propagation (BP) algorithm, where the decoder performs a symbol-wise maximum a posteriori decoding by computing the *a posteriori probabilities* APP for each VN and taking the value that maximizes this quantity. The messages passed are probabilistic information (*beliefs*). The message sent from v to c is the probability that v has a certain value given the observed value and values communicated to v in the previous round. The message from c to v is the probability that v takes a certain value given messages sent to c in the previous round. Usually, the decoding is performed in the Log-Likelihood domain. For a binary random variable X, let $LLR(X) = \frac{P(X=0)}{P(X=1)}$ be the *Log-Likelihood Ratio* (LLR) of X.

The decoder is initialized with the LLRs from the channel, which are received by the n VN. At the beginning of each iteration, each VN takes inputs from the channel and its neighboring CN and sends extrinsic information to its neighboring CN. Then, each CN takes inputs from its neighboring VN and sends extrinsic information to its neighboring VN. After a predetermined stopping criteria has been reached, the decoder computes the LLRs from which the decided bit value is computed.

In this subsection, we present the BP algorithm used in Slepian-Wolf problem: Suppose that a destination wishes to recover a binary i.i.d. sequence X losslessly and has as inputs a noisy version $\hat{Y}$ of X, where we assume that the noise is modeled via a Binary Symmetric Channel

(BSC) of error probability ϵ (BSC-ϵ), and a compressed version S of X that is equal to the syndrome $S = XH^T$.

The following notations will be used:

- $\hat{y}_i$, $i = 1, \ldots, n$: current component values of $\hat{Y}$;
- s_j, $j = 1, \ldots, n-k$: components of the realization of syndrome $S = XH^T$;
- LLR_i: LLR corresponding to channel value $\hat{y}_i$;
- $m_{j,i}^{c \to v}$: LLR message from CN c_j to VN v_i;
- $m_{i,j}^{v \to c}$: LLR message from VN v_i to CN c_j;
- $\mathcal{N}(v)$: set of CN connected to VN v;
- $\mathcal{N}(c)$: set of VN connected to CN c.

The BP algorithm is given in Algorithm 1.

Algorithm 1 Belief propagation for Slepian-Wolf coding

1: Initialization :

$$LLR_i = \log\left(\frac{P(X_i = 0|\hat{y}_i)}{P(X_i = 1|\hat{y}_i)}\right) = (1 - 2\hat{y}_i)\log\frac{1-\epsilon}{\epsilon}$$

2: **for** $l \to iter$ **do**
3: **for** $i \to V_{max}$ **do**
4: Data pass:

$$m_{i,j}^{v \to c} = LLR_i + \sum_{c_{j'} \in \mathcal{N}(v_i) \setminus c_j} m_{j',i}^{c \to v}$$

5: **end for**
6: **for** $j \to C_{max}$ **do**
7: Check pass:

$$m_{j,i}^{c \to v} = 2\tanh^{-1}\left((1 - 2s_j) \prod_{v_{i'} \in \mathcal{N}(c_j) \setminus v_i} \tanh\left(\frac{m_{i',j}^{v \to c}}{2}\right)\right)$$

8: **end for**
9: **for** $i \to V_{max}$ **do**
10: Decision:

$$\hat{x}_i = \begin{cases} 0, \text{ if } LLR_i + \sum_{c_j \in \mathcal{N}(v_i)} m_{j,i}^{c \to v} \geq 0 \\ 1, \text{ else} \end{cases}$$

11: **end for**
12: **end for**

The only step that differs from the BP algorithm used in standard channel decoding is the check pass, where we have to take into account the value of the syndrome that isn't necessary equal to zero.

LDPC are very popular because of their near capacity performance, unfortunately, these can't be achieved with regular LDPC codes and one needs to optimize the degree distribution of the code to come closer to the Shannon bound. One way to perform the optimization is using density evolution and differential evolution.

1.3.2 Optimization and construction

Up so far, we presented a decoding algorithm that can correct some errors, given a LDPC code. One fundamental question is how does the LDPC code choice impact the performance or in other words, given a Tanner graph, how much noise the BP algorithm can correct. Unfortunately this question can't be answered in that form, but we can answer the question for an ensemble of codes, i.e. codes that have the same degree distributions ρ and λ.

Richardson and Urbanke proposed in [Richardson and Urbanke, 2001b] a tool called *density evolution* that determines the asymptotic ensemble average behavior of LDPC codes given the channel model and the noise power. For each iteration of the BP algorithm, this algorithm tracks the average probability density function (pdf) of the messages from VN to CN.

We can define the error probability at iteration l of the BP algorithm as:

$$P_e^{(l)} = \int_{-\infty}^{0} f_v^{(l+1)}(x)dx,$$

where $f_v^{(l+1)}(x)$ is the pdf of the correct message from a VN to a CN during the l-th iteration of BP algorithm. Using this tool, one can determine the decoding threshold of a LDPC code and also optimize its degree distributions using for instance differential evolution.

Definition 1.3.7. *The decoding threshold corresponds to the maximum noise power such that the error probability vanishes as l grows large, $\lim_{l\to\infty} P_e^{(l)} = 0$.*

We present density evolution for the following setup: Assume that the destination wishes to recover a binary i.i.d. sequence X from a noisy version Y, where we assumed that the noise is modeled as a BSC-ϵ.

The following notations will be used:

- $f_v^{(l)}$: pdf of message from a VN to a CN at lth iteration;
- $f_c^{(l)}$: pdf of message from a CN to a VN at lth iteration;
- f_o: pdf of a priori LLR.

Since the path from X to Y is modeled by a BSC-ϵ, f_o is given by

$$f_o = \epsilon\delta\left(x + \log\left(\frac{1-\epsilon}{\epsilon}\right)\right) + (1-\epsilon)\delta\left(x - \log\left(\frac{1-\epsilon}{\epsilon}\right)\right),$$

where $\delta(x)$ is the Dirac distribution.

The average density $f_v^{(l+1)}$ is given by

$$f_v^{(l+1)} = f_o \otimes \left[\sum_{i=1}^{V_{max}} \lambda_i \left(\bigotimes_{k=1}^{i-1} f_c^{(l)}\right)\right],$$

where $\otimes$ stands for the convolution operation and $\bigotimes_{k=1}^{i-1}$ for the convolution of $i-1$ pdfs.

In the same way, the average $f_c^{(l+1)}$ is given by

$$f_c^{(l+1)} = \sum_{i=1}^{C_{max}} \rho_i \Gamma^{-1} \left[\bigotimes_{k=1}^{i-1} \Gamma \left(f_v^{(l+1)} \right) \right],$$

where Γ is the density transformation operator induced by $\gamma : x \to (\mathrm{sgn}(x), -\log\tanh|x/2|)$.

Algorithm 2 describes how density evolution is performed.

Algorithm 2 Density evolution

1: **for** $l \to iter$ **do**
2: $f_v^{(l+1)} = f_o \otimes \left[\sum_{i=1}^{V_{max}} \lambda_i \left(\bigotimes_{k=1}^{i-1} f_c^{(l)} \right) \right]$
3: $f_c^{(l+1)} = \sum_{i=1}^{C_{max}} \rho_i \Gamma^{-1} \left[\bigotimes_{k=1}^{i-1} \Gamma \left(f_v^{(l+1)} \right) \right]$
4: $P_e^{(l)} = \int_{-\infty}^{0} f_v^{(l+1)}(x)dx$
5: **if** $P_e^{(l)} \geq P_e^{(l-1)}$ **then**
6: **break**
7: **end if**
8: **end for**

Differential evolution is a global optimization algorithm that generates, at each iteration l, NP possible degree distributions, denoted by $\{\pi_i^{(l)}\}_{i=1}^{NP}$. Here π denotes a vector grouping a non-redundant set of coefficients of $\lambda(x)$ and $\rho(x)$ satisfying the design rate and other constraints, for instance concentrated check degrees. The first generation, $\{\pi_i^{(0)}\}_{i=1,\ldots,NP}$, is initialized uniformly at random.

At each round, we search within the NP candidates for the one distribution that minimizes the error probability after performing a fixed number of density evolution steps; this distribution is denoted by $\pi_{best}^{(l)}$. For each of the NP distributions, we generate a mutant distribution by randomly selecting four distinct distributions indexed i_1, i_2, i_3, i_4 and computing

$$\pi_i^{(l+1)} = \pi_{best}^{(l)} + F(\pi_{i_1}^{(l)} + \pi_{i_2}^{(l)} - \pi_{i_3}^{(l)} - \pi_{i_4}^{(l)}),$$

where $F > 0$ is a non-negative differential mixing parameter. The NP distributions for the next round are taken to be either the distribution from the previous round or the mutant candidate, if the latter yields a smaller error probability during the density evolution phase.

One major limitation in the BP algorithms performance is the presence of cycles in the Tanner graph.

Definition 1.3.8. *A cycle is defined as a set of connected vertices in the Tanner graph that start and end at the same vertex. The length of a cycle is the number of edges it contains and the girth is defined as the length of the smallest cycle.*

The larger the girth is, the better the code is under BP decoding since the size of the girth is correlated with the number of independent decoding iterations. In iterative decoding, VN and CN send messages in turn along connected edges and outgoing messages are computed from incoming messages. Thus the girth corresponds to the minimal number of half-iterations before (part of) a message can come back to the originating node on a different edge. In other words, since cycles lead to correlations in the marginal probabilities passed by the BP algorithm, the smaller the girth, the larger the effect on the decoding performance.

For long LDPC codes, randomly chosen LDPC codes will always perform well since the concentration theorem assures that the behavior of a randomly chosen code from an ensemble concentrates around the ensemble average. Unfortunately, for simulation, we have to set a code length and no one accepts a code that probably works. Thus, given a code ensemble and a code length, the goal is to find methods to build good codes.

The method considered in this thesis is Progressive Edge Growth (PEG).

PEG algorithm is based on Bit Filling. In both methods, VN are added one at a time. In Bit Filling, edges connecting the new VN are chosen to avoid cycles of size g. For each new VN, a set of feasible CN is computed and the CN chosen within this set is the one that was the less used. In PEG algorithm, edges added are chosen to maximize the local girth at the current VN instead of satisfying a fixed girth requirement.

1.4 Convolutional codes

1.4.1 Definition

Convolutional codes have been introduced by Elias in 1955 and, like LDPC codes presented in the previous section, they have also been used for wireless communications for over 50 years. Again, their strength relies on their easy implemented maximum-likelihood decoder. In opposition to LDPC codes, convolutional codes aren't block codes: the convolutional encoder assigns code bits to an incoming information bit stream, and the output at time t depends not only on the inputs at that time instant but also on previous incoming information bits.

In this thesis we introduce all definitions and concepts based on the convolutional code with generator matrix $[1, 1 + D]$ given on Figure 1.5. We can first note that two coded bits are computed for each information bit, thus the rate of this code is 1/2. We still perform each operation in GF(2), thus the adders are modulo-2 adders. The states of the encoder are given by the content of the memory cells, thus there are two states for this convolutional code: 0 and 1.

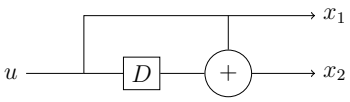


Figure 1.5: A two-state, rate 1/2 convolutional code

We can also note that the coded bits are the result of the convolution of the information bit and two different discrete-time finite-impulse-respond filters with operations in GF(2). The two polynomials are $g_1(D) = 1$ and $g_2(D) = 1 + D$. The encoder outputs the convolution of the input sequence $u(D) = u_0 + u_1 D + u_2 D^2 + \ldots$ by $g_1(D)$ and $g_2(D)$ as $x_i(D) = u(D)g_i(D)$.

Definition 1.4.1. *The matrix $G(D) = [g_1(D) \quad g_2(D)]$ is called the code's generator matrix and the polynomials $g_i(D)$ are the generator polynomials. As for LDPC codes, we can also use a parity check matrix $H(D)$ to describe a convolutional code and $H(D)$ is such that $G(D)H(D) = 0$.*

In the previous section, we saw that the Tanner graph, which is a graphical representation of LDPC codes, is very useful for the decoding procedure: trellis representation is the analog for convolutional codes. The trellis gives transitions, labeled by the coded bits, between the encoder states, for a given input information bit.

For our example code, one trellis section is given on Figure 1.6. Transitions are labeled by the output values x_1x_2, dashed lines correspond to $u = 0$ and solid lines to $u = 1$.

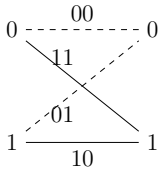


Figure 1.6: One trellis section of the convolutional code with generator matrix $[1, 1 + D]$

There exists many different decoders for convolutional codes. The most famous two are the Viterbi and the BCJR algorithm. The Viterbi algorithm is a maximum-likelihood sequence decoder, thus it minimizes the probability of code-sequence error, whereas the BCJR is a bit-wise maximum a posteriori decoder, which minimizes the probability of information bit error.

For the rest of the section, we assume that the coded sequence is of length K and denote by $\underline{y}_{1\to K}$ the received sequence.

1.4.2 Algorithms

The Viterbi algorithm [Viterbi, April 1967] is a dynamic programming maximum-likelihood decoding procedure that updates a set of trellis vertices (states) at every time unit. At every time unit, it computes the *path metric* for all paths entering a trellis state using the metrics at the previous time unit and the received sequence. This path metric is the cumulated Hamming distance between the received sequence and a given path in the trellis. It then compares all incoming metrics of a state and chooses the path with the smallest metric, or uses a tie-breaking rule when more than one path has minimal metric. Thus, at the end of the decoding, the codeword that is closest to the received sequence is output.

Let Λ_k denotes the cumulated metric vector at time unit k. In our case this vector has two components: the cumulated metric of state 0 and the one of state 1. Let y_1y_2 denote the received sequence at time unit k. The cumulated metric vector for the next time unit is given by:

$$\Lambda_{k+1} = \begin{bmatrix} \min\{\Lambda_k(0) + d_H(y_1y_2, 00); \Lambda_k(1) + d_H(y_1y_2, 01)\} \\ \min\{\Lambda_k(0) + d_H(y_1y_2, 11); \Lambda_k(1) + d_H(y_1y_2, 10)\} \end{bmatrix}.$$

The Viterbi algorithm is given in Algorithm 3.

Algorithm 3 Viterbi algorithm

1: Initialization :
$$\Lambda_0 = [0; 1]^T$$
2: **for** $k \to K$ **do**
3: Compute the new cumulated metric vector Λ_{k+1} and for each component j of Λ_{k+1} store the path minimizing $\Lambda_{k+1}(j)$.
4: **end for**
5: Decision: Choose the path associated with the minimal component of the cumulated metric vector

The BCJR algorithm, proposed by Bahl, Cocke, Jelinek and Raviv in [Bahl et al., Mar. 1974] is a maximum a posteriori decoder, the goal is thus to compute the APP $P(u_k|\underline{y}_{1\to K})$.

First note that $P(u_k, \underline{y}_{1\to K}) = P(u_k|\underline{y}_{1\to K})P(\underline{y}_{1\to K})$, since the second term is a constant, the searched APPs are proportional to $P(u_k, \underline{y}_{1\to K})$.

Let $\underline{s}_k$ denote the content of the memory cells at time unit k and x_k the output vector at time unit k.

$$\begin{aligned} p(u_k, \underline{y}_{1\to K}) &= \sum_{\underline{s}_k} \sum_{\underline{s}_{k-1}} \sum_{x_k} P(u_k, \underline{s}_{k-1}, \underline{s}_k, x_k, \underline{y}_{1\to K}) \\ &\propto \sum_{\underline{s}_k} \sum_{\underline{s}_{k-1}} \sum_{x_k} \alpha_{k-1}(\underline{s}_{k-1})\beta_k(\underline{s}_k)\lambda(x_k)P(x_k|\underline{s}_k, \underline{s}_{k-1})P(\underline{s}_k|\underline{s}_{k-1}, u_k) \end{aligned} \tag{1.3}$$

where

$$\begin{aligned} \alpha_{k-1}(\underline{s}_{k-1}) &= P(\underline{y}_{1\to k-1}|\underline{s}_{k-1}), \\ \beta_k(\underline{s}_k) &= P(\underline{y}_{k+1\to K}|\underline{s}_k) \text{ and} \\ \lambda(x_k) &= P(y_k|x_k). \end{aligned}$$

Let $V_p(\underline{s}_k)$ be defined as $V_p(\underline{s}_k) = \{\underline{s}_{k-1}|P(\underline{s}_k|\underline{s}_{k-1}) \neq 0\}$ and $V_f(\underline{s}_{k-1})$ as $V_f(\underline{s}_{k-1}) = \{\underline{s}_k|P(\underline{s}_k|\underline{s}_{k-1}) \neq 0\}$. One can prove that

$$\forall k \in \{1, \ldots, K\}, \quad \alpha_k(\underline{s}_k) = \sum_{\underline{s}_{k-1} \in V_p(\underline{s}_k)} \alpha_{k-1}(\underline{s}_{k-1})\lambda(x_k) \tag{1.4}$$

and

$$\forall k \in \{1, \ldots, K\}, \quad \beta_{k-1}(\underline{s}_{k-1}) = \sum_{\underline{s}_k \in V_f(\underline{s}_{k-1})} \beta_k(\underline{s}_k)\lambda(x_k). \tag{1.5}$$

Thus, the BCJR algorithm is a forward/backward decoding algorithm: the α metric being evaluated in a forward fashion on the trellis and the β metric in a backward fashion.

The BCJR algorithm is given in Algorithm 4.

Algorithm 4 BCJR algorithm

1: Initialization: Compute all $\lambda(x_k)$, set $\alpha_0(s_0) = [1; 0; \ldots; 0]^T$ and $\beta_K(s_K) = [1; 0; \ldots; 0]^T$.
2: **for** $k = 1 \to K$ **do**
3: Compute recursively the α metric using (1.4)
4: **end for**
5: **for** $k = K \to 1$ **do**
6: Compute recursively the β metric using (1.5)
7: **end for**
8: **for** $k = 1 \to K$ **do**
9: Compute $P(u_k, \underline{y}_{1\to K})$ using (1.3) and decide $\hat{u}_k = \arg\max P(u_k, \underline{y}_{1\to K})$.
10: **end for**

Convolutional codes can be used for quantization: The codeword $\underline{\hat{y}}$ closest in Hamming metric to the input sequence $\underline{y}_{1\to K}$ is found using the Viterbi algorithm and the corresponding information sequence is output as index w.

Chapter 2

Fundamental Hamming-space Voronoi region of a convolutional code

This work was presented in [Savard and Weidmann, 2015c].

Contents

As mentioned in Chapter 1, the improved decoder that will be presented in Chapter 3 relies on the characterization of all the Voronoi cells $\mathcal{V}_w$ of the convolutional code used in the side information branch. Since convolutional codes are linear, a characterization of the fundamental Hamming-space Voronoi region $\mathcal{V}_0$ of this convolutional code is sufficient.

We describe the characterization of the Voronoi cell $\mathcal{V}_0$ of a convolutional code in a tutorial way based on the rate 1/2 convolutional code with generator matrix $[1, 1 + D]$ presented in the previous chapter. This characterization can of course be generalized to any convolutional code.

The encoder of a convolutional code can be represented by a Finite State Machine (FSM) and as seen previously, its trellis representation allows an efficient decoding using the Viterbi or the BCJR algorithm.

The performance of a convolutional code may be evaluated by computing the decoder error probability. Works based on the union bound and the distance spectrum first gave useful approximations. First works on the exact error performance computation have been investigated by Morrissey using a Markov chain describing the decoder operation [Morrissey, July 1970]. He extended this work to the Viterbi decoding algorithm for a class of convolutional codes [Morrissey, Oct. 1969] and [Best et al., March 1995] later generalized this approach to any convolutional code. All these methods were based on this FSM approach.

[Calderbank et al., May 1995] used the same Markov chain approach to evaluate the quantization performance of binary convolutional codes. Using this formalism, they were able to derive exact formulas for the expected incremental Mean Square Error (MSE) per dimension for arbitrarily long sequences. The maximum incremental MSE may be seen as the covering radius of the convolutional code. Using their Markov chain characterization, one can more easily derive exact formulas for the expected performance, which is rarely possible, due to computational difficulties.

When quantizing uniformly distributed sources, the Markov chain describes a uniform distribution over all Voronoi cells of the convolutional code. In this chapter, we introduce a characterization of the fundamental Hamming-space Voronoi region $\mathcal{V}_0$ of a convolutional code, that is the Voronoi cell of the all-zero codeword.

[Agrell, Jan. 1996] proposed a characterization of the Voronoi cells of linear block codes in Euclidean metric, which is relevant to soft decoding. Here, we provide a similar characterization, but for convolutional decoding using Hamming distance as metric. In terms of the standard array of a code, we search for the first column of the table, i.e. all coset leaders (which are infinite sequences for convolutional codes).

The fundamental Hamming-space Voronoi cell $\mathcal{V}_0$ is the set of all sequences that are closer, in terms of Hamming distance, to the all-zero codeword than to any other codeword (ties can be broken arbitrarily). Since the code is linear, the goal is to write the space of all sequences $\mathcal{E}$ as the direct sum of the Voronoi cells $\mathcal{V}_c$ associated to each codeword c. The cells $\mathcal{V}_i$ are such that $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$.

$$\mathcal{E} = \bigcup_{c \in \mathcal{C}} \mathcal{V}_c = \bigcup_{c \in \mathcal{C}} \mathcal{V}_0 \oplus c, \tag{2.1}$$

where $\mathcal{C}$ is the code and $\oplus$ denotes componentwise modulo-2 addition.

Our problem is equivalent to enumerating all channel sequences that the Viterbi algorithm decodes to the all-zero codeword. When we perform our characterization, we must consider a tie-breaking rule in the Viterbi decoding algorithm such that the constraint (2.1) is satisfied. Instead of a deterministic rule, we will use a probabilistic one, resulting in a simpler Markov chain characterization.

The rest of the chapter is organized as follows: Section 2.1 present the convolutional decoder FSM, Section 2.2 explains how to get the FSM of the Voronoi cell $\mathcal{V}_0$ using this convolutional decoder FSM and finally, Section 2.3 gives the transition probabilities of this new obtained FSM.

2.1 Convolutional decoder FSM

States in the encoder state diagram of the convolutional code (the content of the memory cell) are called *encoder states* or *trellis states*. For example, the convolutional code with generator matrix $[1, 1 + D]$ has two encoder states: state 0 and state 1.

The internal states of the Viterbi decoder, i.e. the path metrics for each encoder state, are called *decoder states*. For the convolution code with generator matrix $[1, 1 + D]$, the decoder states are pairs of individual path metrics. A possible decoder state for our example is $[1, 2]$.

Since the Viterbi algorithm computes the new metric using the previous metrics and the sequence received from the channel, the choices made by the algorithm depend only on the metric differences. Thus we can form *metric states* that are normalized decoder states where the minimum component has been subtracted. The minimal component of every metric state (vector) will be 0.

For the above example, the metric state is $[0, 1]$.

Since the set of possible path metrics with this normalization step is finite (the starting state of the algorithm is known), the number of metric states is also finite. For the code used, the Hamming distance between a label and the received sequence is at most 2, thus the set of possible metric states is $\{[0, 0], [0, 1], [1, 0], [0, 2], [2, 0]\}$.

Definition 2.1.1. *A probabilistic finite state machine is a tuple* $(\mathcal{S}, \Sigma, \mathcal{T}, P, s_0, \mathcal{F})$ *where*

- $\mathcal{S}$ *is a finite set of states;*

- Σ *is the label alphabet;*
- $\mathcal{T} \subseteq \mathcal{S} \times \Sigma \times \mathcal{S}$ *is the set of transitions, where for each transition* $e \in \mathcal{T}$, $\sigma(e)$ *denotes its starting state,* $\tau(e)$ *its ending state and* $\underline{o}(e)$ *its label;*
- $P(e), e \in \mathcal{T}$ *are the transition probabilities;*
- $s_0 \in \mathcal{S}$ *is the initial state;*
- $\mathcal{F} \subseteq \mathcal{S}$ *is the set of proper final states.*

Thus, for our example, the FSM of the Viterbi decoder (denoted FSM-dec) is characterized by:

- $\mathcal{S} = \{[0,0],[0,1],[1,0],[0,2],[2,0]\}$;
- $\Sigma = \{00, 01, 10, 11\}$;
- $\mathcal{T}$ is given in the table of Figure 2.1;
- P will depend on the channel model;
- $s_0 = [0,2]$[1];
- $\mathcal{F} = \mathcal{S}$: all states are final states[2].

A graphical representation of FSM-dec is given in Figure 2.1.

This FSM can be used to compute the exact error probability of the decoder, as done in [Best et al., March 1995] or [Calderbank et al., May 1995].

2.2 FSM of the Voronoi cell $\mathcal{V}_0$ of a convolutional code

2.2.1 FSM of the Voronoi cell $\mathcal{V}_0$ of a convolutional code using the Viterbi decoding algorithm

Since we are interested in the characterization of the fundamental Hamming-space Voronoi cell $\mathcal{V}_0$, we must build a FSM that only keeps metric states such that the sequence is decoded onto the all-zero codeword (using the Viterbi algorithm). This will be denoted by FSM-$\mathcal{V}_0$.

Proposition 2.2.1. *A necessary condition for a metric state* $\underline{m} = [m_0, \ldots, m_{|S_e|-1}] \in \mathcal{M}$ *to be kept in FSM-*$\mathcal{V}_0 \subseteq$ *FSM-dec is given by:*

$$\min_{e \in \mathcal{T}: \tau(e)=0} \{m_{\sigma(e)} + d_H(\underline{r}, \underline{o}(e))\} = m_0 + d_H(\underline{r}, \underline{0}). \tag{2.2}$$

S_e *is the set of encoder states,* $\underline{r}$ *the received sequence and* $d_H(x,y)$ *the Hamming distance between* x *and* y.

[1] The Viterbi decoder starts in state 0 with metric 0, while state 1 can be thought of having metric ∞. However, it can be shown that starting in $[0,2]$ yields the same behavior.

[2] For channel coding, the code is generally terminated in a specific state, whereas this is not necessary for quantization. This will of course impact the way we treat termination for our FSM.

	Received sequence			
Metric state	00	01	10	11
$[0,0]$	$[0,1]$	$[0,1]$	$[1,0]$	$[1,0]$
$[0,1]$	$[0,2]$	$[0,0]$	$[0,0]$	$[2,0]$
$[1,0]$	$[0,0]$	$[0,2]$	$[2,0]$	$[0,0]$
$[0,2]^*$	$[0,2]$	$[0,0]$	$[0,0]$	$[2,0]$
$[2,0]$	$[0,0]$	$[0,2]$	$[2,0]$	$[0,0]$

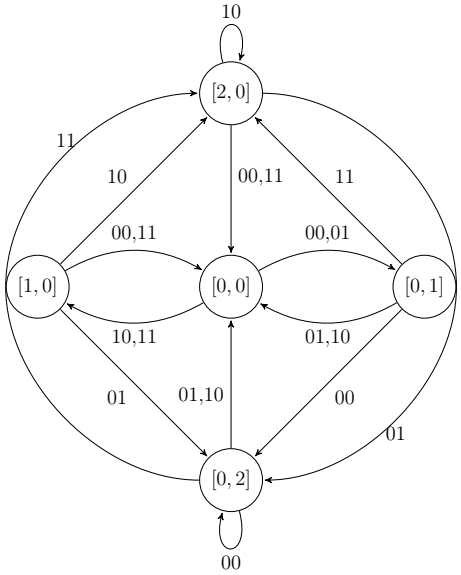


Figure 2.1: Transition table of FSM-dec and its graphical representation for the convolutional code with generator matrix $[1, 1+D]$ (The metric state with the asterisk in the table corresponds to the initial state.)

Proof. The proposition can be rephrased as follows: among all transitions (edges) entering state 0, the transition with the all-zero label coming from state 0 must have the smallest Hamming distance from the received sequence. This necessary condition assures that the sequence may be mapped onto the all zero codeword, and thus lie in $\mathcal{V}_0$. □

In our example, we have two transitions entering state 0: one coming from state 0 with label $\underline{o}_{0\to 0} = 00$ and one coming from state 1 with label $\underline{o}_{1\to 0} = 01$. Suppose that the metric state at time i is $[m_0(i), m_1(i)]^T$ and that the received sequence at time $i+1$ is $r_1 r_2$. The new metrics are given by

$$\begin{bmatrix} m_0(i+1) \\ m_1(i+1) \end{bmatrix} = \begin{bmatrix} \min\{m_0(i) + d_H(r_1 r_2, 00), m_1(i) + d_H(r_1 r_2, 01)\} \\ \min\{m_0(i) + d_H(r_1 r_2, 11), m_1(i) + d_H(r_1 r_2, 10)\} \end{bmatrix}. \tag{2.3}$$

We only keep the metric state:

$$\begin{bmatrix} m_0(i+1) - \min\{m_0(i+1), m_1(i+1)\} \\ m_1(i+1) - \min\{m_0(i+1), m_1(i+1)\} \end{bmatrix} = \begin{bmatrix} \max\{0, m_0(i+1) - m_1(i+1)\} \\ \max\{0, m_1(i+1) - m_0(i+1)\} \end{bmatrix} \tag{2.4}$$

if the following equality holds

$$\min\{m_0(i) + d_H(r_1 r_2, \underline{o}_{0\to 0}), m_1(i) + d_H(r_1 r_2, \underline{o}_{1\to 0})\} = m_0(i) + d_H(r_1 r_2, 00), \tag{2.5}$$

which insures that we stay in the Voronoi cell $\mathcal{V}_0$.

Transitions from the previously obtained FSM-dec that do not satisfy this necessary condition are removed, yielding a new FSM given in Table 2.1.

We notice that metric state $[2, 0]$ has no transitions left to other metric states, thus we can remove it too, yielding the Voronoi FSM in Figure 2.2. This removal procedure may need to be applied repeatedly.

	Received sequence			
Metric state	00	01	10	11
$[0,0]$	$[0,1]$	–	$[1,0]$	–
$[0,1]$	$[0,2]$	$[0,0]$	$[0,0]$	$[2,0]$
$[1,0]$	$[0,0]$	–	$[2,0]$	–
$[0,2]^*$	$[0,2]$	$[0,0]$	$[0,0]$	$[2,0]$
$[2,0]$	–	–	–	–

Table 2.1: FSM-$\mathcal{V}_0$ of the Voronoi cell $\mathcal{V}_0$ for the convolutional code with generator matrix $[1, 1+D]$

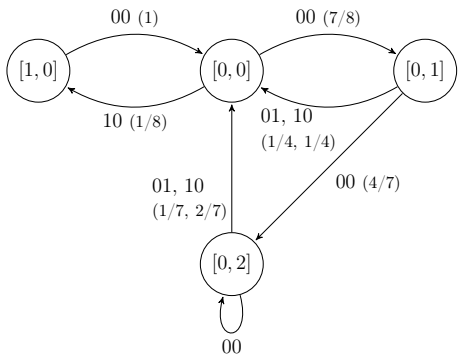


Figure 2.2: FSM-$\mathcal{V}_0$ of the Voronoi cell $\mathcal{V}_0$ for the convolutional code with generator matrix $[1, 1+D]$. In parentheses: branch transition probabilities

Closer inspection of the metric states reveals that they fall into two classes: $\mathcal{F}$, which contains the metric states in which the metric of state 0 is the smallest one, and $\bar{\mathcal{F}}$, containing the other metric states. The class $\mathcal{F}$ contains the proper final states, that are the states in which a finite-length sequence may terminate and be guaranteed to lie in $\mathcal{V}_0$.

For the same reason, the finite-length decoder can't stop in a metric state of class $\bar{\mathcal{F}}$ since the all-zero codeword won't be decoded. Nevertheless, a decoding trajectory can go through metric states of that second class and still be mapped onto the all-zero codeword. For instance, the sequence 0010100000, corresponding to the trajectory $[0,2] \to [0,2] \to [0,0] \to [1,0] \to [0,0] \to [0,1]$, is indeed mapped to the all-zero codeword using the Viterbi algorithm.

The characterization of $\mathcal{V}_0$ must take this particularity into account. If we consider finite-length sequences, the trellis built from this FSM must only keep metric states of class $\mathcal{F}$ in the last stage.

Thus, for our example, FSM-$\mathcal{V}_0$ is characterized by:

- $\mathcal{S} = \{[0,0], [0,1], [1,0], [0,2]\}$;
- $\Sigma = \{00, 01, 10\}$
- $\mathcal{T}$ is given in Table 2.1;
- $s_0 = [0,2]$;
- $\mathcal{F} = \{[0,0], [0,1], [0,2]\}$.

A graphical representation of FSM-$\mathcal{V}_0$ is given in Figure 2.2.

2.2.2 FSM of the Voronoi cell $\mathcal{V}_0$ of a convolutional code using the syndrome decoding algorithm

[Schalkwijk et al., September 1978] proposed another approach to decode convolutional codes: the *syndrome decoding algorithm*. In [Schalkwijk et al., November 1979], they proposed a new characterization of the decoder error probability using this algorithm. It can be also used to characterize the cell $\mathcal{V}_0$. For this ML decoder, the decoder first forms the syndrome $\underline{z}$ and then, using a recursive algorithm like Viterbi's, a tuple of noise sequence of minimum Hamming weight that may be a possible cause for this syndrome is output. Using this noise sequence estimate, an estimate of the sent message can be form.

Assume that the data sequence is $u(D) = u_0 + u_1 D + u_2 D^2 + \dots$ and that we use the convolutional whose generator polynomials are $g_1(D) = 1$ and $g_2(D) = 1 + D$. We denote the encoded outputs as $x_i(D) = u(D)g_i(D), i = \{1, 2\}$. The noisy outputs are defined as $y_i(D) = x_i(D) + n_i(D)$. The syndrome is obtained by

$$\begin{aligned} z(D) &= g_2(D)(g_1(D)u(D) + n_1(D)) + g_1(D)(g_2(D)u(D) + n_2(D)) \\ &= g_2(D)n_1(D) + g_1(D)n_2(D). \end{aligned}$$

Using a recursive algorithm, the noise sequence $\hat{n}_1(D)$, $\hat{n}_2(D)$ of minimum weight that could cause this syndrome is obtained. The estimate data sequence is computed as

$$\hat{x}_i(D) = y_i(D) + \hat{n}_i(D).$$

The syndrome former of the convolutional code with generator matrix $[1, 1 + D]$ is given on Figure 2.3. One section of the syndrome former trellis is also given on the same figure. Each transition of the trellis is labeled with the error bit $[n_1, n_2]$. Transitions marked with dashed lines correspond to $z = 0$ and solid line to $z = 1$.

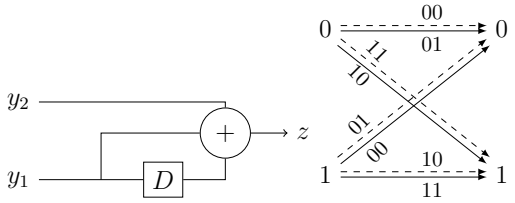


Figure 2.3: Syndrome former and one trellis section for the convolutional code with generator matrix $[1, 1 + D]$: dashed lines correspond to $z = 0$ and solid line to $z = 1$.

Example 2.2.1. *Assume that* $u = 0100$. *Thus* $x_1 = 0100$ *and* $x_2 = 0110$. *After transmission, the received sequence is given by* $y_1 = 0101$ *and* $y_2 = 1110$, *thus the syndrome is given by* $z = 1001$. *Using the Viterbi algorithm on the trellis given on Figure 2.3, one obtains* $\hat{n}_1 = 0001$ *and* $\hat{n}_2 = 1000$ *and thus* $\hat{x}_1 = 0100$ *and* $\hat{x}_2 = 0110$.

Since the cell $\mathcal{V}_0$ is composed of all coset-leaders, in other words all the sequences of minimum Hamming weight associated with all possible syndromes, we can again form a FSM based on other metric states, given on Table 2.2. Using this FSM, we can form all sequences contained in $\mathcal{V}_0$.

2.3 Transition probabilities in the Voronoi FSM

The goal of assigning probabilities to the transitions of FSM-$\mathcal{V}_0$ is to obtain a uniform probability distribution over all sequences in $\mathcal{V}_0$. The equivalent problem for FSM-dec is trivial, since

	$z=0$		$z=1$	
Metric state	Metric state	Survivor	Metric state	Survivor
$[0,/]$	$[0,2]$	$0,0$	$[0,0]$	$0,0$
$[0,2]$	$[0,2]$	$0,0$	$[0,0]$	$0,0$
$[0,0]$	$[0,1]$	$0,1$	$[0,1]$	$1,0$
$[0,1]$	$[0,2]$	$0,(0,1)$	$[0,0]$	$(0,1),0$

Table 2.2: FSM of the Voronoi cell $\mathcal{V}_0$ for the convolutional code with generator matrix $[1, 1+D]$ using the syndrome decoding algorithm. Columns 3 and 5 list the survivor (i.e. the encoder state with minimum weight transition): if there is a choice of survivor, candidates are given within parentheses.

uniform probabilities over the 2^n transitions leaving every metric state (for rate-k/n codes) will result in uniform probability over the space of binary sequences. This does not hold for FSM-$\mathcal{V}_0$, since some outgoing transitions may not always be available due to tie-breaking in the Viterbi algorithm. The correct probabilities can be found using a weighted adjacency matrix.

Proposition 2.3.1. *Let the weighted adjacency matrix $A = [a_{i,j}]$, where*

$$a_{i,j} = \sum_{e\in\mathcal{T}:\substack{\sigma(e)=i\\ \tau(e)=j}} \frac{1}{w_0(e)}$$

and $w_0(e)$ is the number of minimum-metric branches entering Viterbi decoder state 0 in metric state j. Then the metric state transition probabilities are obtained from the positive eigenvector $\underline{\mu}_0$ corresponding to the Perron-Frobenius eigenvalue λ_0 of A as follows:

$$p_{i,j} = \Pr\{s_t{=}j | s_{t-1}{=}i\} = \frac{a_{i,j}\mu_{0,j}}{\lambda_0\mu_{0,i}}. \tag{2.6}$$

The probabilities of individual branches e such that $\sigma(e) = i$ and $\tau(e) = j$ is $Pr(e) = \frac{p_{i,j}}{a_{i,j}w_o(e)}$ (so their sum is $p_{i,j}$ as expected). The rate of FSM-$\mathcal{V}_0$ equals $\frac{1}{n}\log_2\lambda_0$.

Proof. The proof relies on counting the number of sequences in $\mathcal{V}_0$ and computing its asymptotic growth rate. The number of sequences following a specific path on FSM-$\mathcal{V}_0$ is given by the product of the $a_{i,j}$ along the path. Thus $a_{i,j}$ is incremented by $1/w_0(e)$ for each transition e leading from metric state i to j. The division by $w_0(e)$ accounts for the fact that the tie-breaking rule in the Viterbi algorithm picks one out of the $w_0(e)$ winning paths entering state 0. (A tie-breaking rule not respecting this proportion would result in $\mathcal{V}_0$ being either too small or too large.)

Let $N_u^t(i)$ be the number of sequences of length $(t-u)n$ bits beginning in metric state i at time u. (One time step of the FSM-$\mathcal{V}_0$ corresponds to n bits.) Then one has $N_0^t(i) = \sum_j a_{i,j}N_1^t(j)$, or $\underline{N}_0^t = A\underline{N}_1^t$ using column vectors. For $t\to\infty$, the asymptotic growth is dominated by the Perron-Frobenius eigenvalue λ_0, i.e. $\underline{N}_0^t = \lambda_0^t\underline{\mu}_0$, obtained from the equation $A\underline{\mu} = \lambda\underline{\mu}$. The rate per output bit of FSM-$\mathcal{V}_0$ is thus $\frac{1}{n}\log_2\lambda_0$.

The transition probabilities $p_{i,j}$ are well-defined, since

$$\sum_j p_{i,j} = \sum_j \frac{a_{i,j}\mu_{0,j}}{\lambda_0\mu_{0,i}} = \frac{[A\underline{\mu}_0]_i}{\lambda_0\mu_{0,i}} = 1. \tag{2.7}$$

To see that these are the correct $p_{i,j}$, one may compute the rate R of the probabilistic FSM-$\mathcal{V}_0$ and compare with the rate defined using the eigenvalue λ_0. Notice that R will be larger than the entropy rate of the underlying Markov chain, since there are in general multiple distinctly labeled weighted transitions between states. Each bundle of parallel edges from state s_i to s_j contributes $P(s_i \to s_j) \log a_{i,j}$ to the rate (we will explain below why sometimes this does not coincide with the edge label entropy). Let the row vector $\underline{\pi}$ be the stationary distribution of $P = [p_{i,j}]$, obtained from $\underline{\pi} P = \underline{\pi}$. Then

$$\begin{aligned} R &= -\sum_i \pi_i \left[H(S_t | S_{t-1} = i) + \sum_j p_{i,j} \log a_{i,j} \right] \\ &= \sum_i \pi_i \left[H(p_{i,1}, p_{i,2}, \ldots) + \sum_j p_{i,j} \log a_{i,j} \right] \end{aligned}$$

where the first term in brackets is the Markov chain conditional entropy. One has

$$\begin{aligned} R &= \sum_i \pi_i \left[H(p_{i,1}, p_{i,2}, \ldots) + \sum_j p_{i,j} \log a_{i,j} \right] \\ &= \sum_i \pi_i \left[-\sum_j p_{i,j} \log \frac{a_{i,j} \mu_{0,j}}{\lambda_0 \mu_{0,i}} + \sum_j p_{i,j} \log a_{i,j} \right] \\ &= \log \lambda_0 + \sum_i \pi_i \log \mu_{0,i} - \sum_i \pi_i \sum_j p_{i,j} \log \mu_{0,j} \\ &= \log \lambda_0, \end{aligned}$$

where the last equality is due to the eigenvalue equation $\sum_i \pi_i p_{i,j} = \pi_j$ (see problem 6.13 in [Ash, 1990]). □

The rate computation in the proof reflects a peculiarity in the probabilistic FSM model for $\mathcal{V}_0$, namely the fact that $a_{i,j}$ may be a sum of non-integer fractions. In that case the edge label entropy will be larger than $\log a_{i,j}$. For example, for two parallel edges with $w_0 = 2$ one has $H = 1$, but $\log a_{i,j} = \log(1/2 + 1/2) = 0$. The reason of this discrepancy is that the probabilistic model is insufficient to capture the actual tie-breaking behavior. This will be illustrated in the following.

Example 2.3.1. *Consider the two-state FSM with*

$$\mathcal{T} = \left(s_0 \xrightarrow[w_0=1]{a} s_1, s_0 \xrightarrow[w_0=2]{b} s_1, s_1 \xrightarrow[w_0=1]{c} s_0, s_1 \xrightarrow[w_0=1]{d} s_0 \right),$$

where a, b, c, d stand for distinct output symbols. It has $a_{0,1} = 1.5$, $a_{1,0} = 2$ and thus $\lambda_0 = \sqrt{3}$. The symbols a and b will be output in proportion $p_a/p_b = 2/1$ and so the apparent output entropy on the transition $s_0 \to s_1$ is $h(1/3) \approx 0.92 > \log a_{0,1} \approx 0.58$. Accounting for a proper tie-breaking rule requires introducing additional states to remove this spurious uncertainty (entropy). A necessary and sufficient condition for this is that all transitions in the new FSM have unit weight, i.e. $w_0 = 1$. This is easily done for the present example by extending the original FSM with a state s_2 as follows:

$$\mathcal{T}' = \left(s_0 \xrightarrow[w_0=1]{a} s_1, s_0 \xrightarrow[w_0=1]{b} s_2, s_1 \xrightarrow[w_0=1]{c} s_0, s_1 \xrightarrow[w_0=1]{d} s_0, s_2 \xrightarrow[w_0=1]{c} s_0, \right),$$

that is, the tie-breaking rule says that b will always be followed by c (one could have chosen d instead of c, but this choice must be deterministic). The point is that tie-breaking must guarantee that a appears exactly *twice as often as b, rather than on average. The original FSM is seen to be a projection of the extended FSM in which s_1 and s_2 are superposed, leading to the spurious probabilistic choice that increases entropy. In other words, the original FSM does not have sufficient memory to correctly model tie-breaking.*

It is easily seen that Markov chain output entropy and actual rate coincide if and only if tie-breaking is such that $w_0(e) = 1$, $\forall e \in \mathcal{T}$. Indeed, for a state transition $s_i \to s_j$ we have the entropy due to parallel edges

$$\begin{aligned} H_{i,j} &= -\sum_{e \in \mathcal{T}_{i,j}} \frac{1}{a_{i,j} w_0(e)} \log \frac{1}{a_{i,j} w_0(e)} \\ &= \log a_{i,j} + \sum_{e \in \mathcal{T}_{i,j}} \log w_0(e) \geq \log a_{i,j}, \end{aligned}$$

with equality if and only if $w_0(e) = 1$, $\forall e \in \mathcal{T}_{i,j}$.

The tie-breaking rule must be deterministic in the sense that out of all prefix sequences leading to a tie, only a given fraction may be extended further. It is thus not sufficient that the model assigns smaller probabilities to the excluded prefixes; it needs to have enough memory to prevent those prefixes from appearing altogether.

The analysis of average quantizer distortion in [Calderbank et al., May 1995] did not require considering tie-breaking rules, because the average is computed over all paths through the trellis, i.e. over all Voronoi cells at the same time. This works even in the case of non-uniform source probabilities. However, tie-breaking needs to be taken into account when computing the average number of information bit errors of a Viterbi channel decoder, as pointed out in [Best et al., March 1995]. In that work, the authors compared different rules that could be analyzed using a simple decoder FSM alone, and provided (numerical) approximations for more complex decoders.

It turns out that FSM-$\mathcal{V}_0$ as constructed above from a Viterbi decoder with probabilistic (approximate) tie-breaking is precise enough to yield remarkable gains in applications, as will be shown in Chapter 3. Nevertheless it is straightforward to obtain an exact model of $\mathcal{V}_0$: it suffices to start from an FSM model of a Viterbi decoder that includes a tie-breaking rule and then applying Proposition 2.2.1 and Proposition 2.3.1, since the tie-breaking will ensure $w_0(e) = 1$ for all transitions. The above example did this "backwards," without constructing the full decoder FSM.

Chapter 3

Improved practical scheme

This work was presented in [Savard and Weidmann, 2013a], [Savard and Weidmann, 2013b] and [Savard and Weidmann, 2014a].

Contents

In this chapter, we propose to use the characterization of $\mathcal{V}_0$, presented in Chapter 2, to improve an LDPC-based decoder for source coding with coded SI , presented in Chapter 1. We first present how to improve the standard decoder and then how to perform a numerical density evolution, leading to code optimization.

3.1 Geometrical intuition and decoding procedure

Our method relies on the following observation: the coded SI is given by a single index w, but many SI sequences are mapped onto the same index by the quantizer. The set of sequences mapped onto the index w forms the Voronoi cell $\mathcal{V}_w = \{\underline{y} \in \{0,1\}^n | E_Y(\underline{y}) = w\}$. Thus all sequences within $\mathcal{V}_w$ could have been possible $\underline{y}$ sequences. Instead of using only $\hat{\underline{y}}(w)$ to decode $\underline{x}$, our method exploits the knowledge of $\mathcal{V}_w$ to select another representative sequence. This new sequence will be obtained by projecting onto $\mathcal{V}_w$ an approximate solution $\hat{\underline{x}}^{(t)}$, obtained at some iteration t by the BP decoder. The hope is that this new sequence is closer to $\underline{x}$ than $\hat{\underline{y}}(w)$ and thus accelerates decoding. The device that performs this projection and outputs a new SI sequence $\hat{\underline{y}}'$ (with reliability information) will be called a *Voronoi decoder*.

The decoding is done as follows: Decoding of X starts with T BP LDPC decoder iterations as in the standard setup, using the LLR associated to $\hat{Y}(w)$ as channel values. If the decoder fails to converge after T iterations, the SI is modified by performing a Voronoi projection. The resulting sequence is used to modify the input LLRs before carrying on with additional BP iterations (the set of messages from CN to VN and from VN to CN is not reset). If the decoder still fails to converge after t additional decoding iterations, the above procedure is restarted. The geometrical intuition behind this decoding principle is depicted on Figure 3.1, where three projection steps are performed.

Figure 3.2 depicts the decoder graph, where the Voronoi decoder is responsible for modifying the channel LLR values fed to the LDPC decoder graph above it.

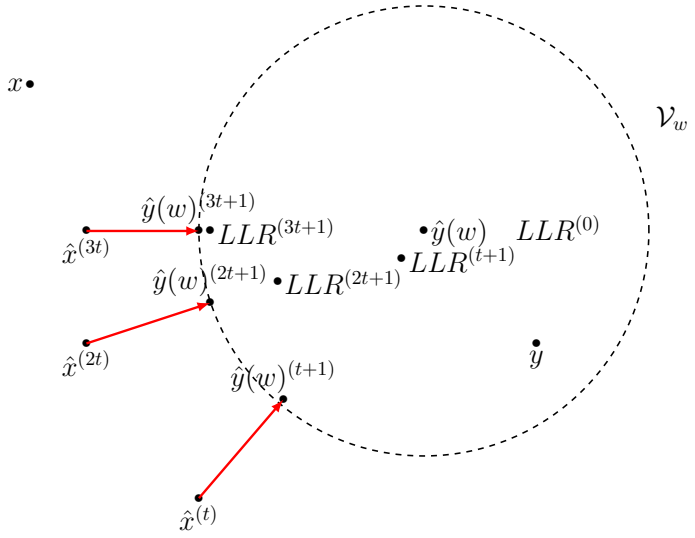


Figure 3.1: Geometrical intuition: 3 steps of the proposed algorithm

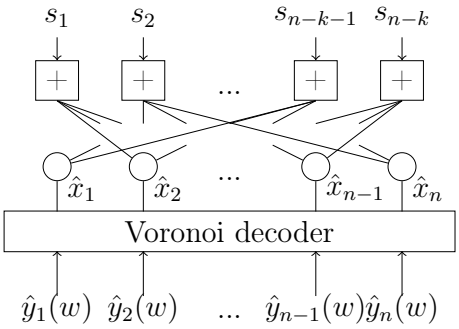


Figure 3.2: Proposed decoder graph

Thus, our improved method relies on a characterization of all the Voronoi cells $\mathcal{V}_w$ of the convolutional code used in the SI branch. Because convolutional codes are linear, the characterization of the Voronoi cell $\mathcal{V}_0$ associated to the all-zero codeword is sufficient, since other cells are obtained as $\mathcal{V}_w = \mathcal{V}_0 \oplus \hat{Y}(w)$, where $\oplus$ denotes component-wise modulo-2 addition. This characterization is done by building the FSM associated with $\mathcal{V}_0$ as explained in the previous chapter.

3.2 Different implementations of the Voronoi decoder

We focused on two Voronoi decoder implementations: on one hand an hard-input, hard-output Viterbi-based decoder and on the other hand a soft-input, soft-output BCJR-based decoder.

For both decoders, the input is an estimate $\underline{\hat{x}}^{(t)}$ of the source sequence, whose reliability is crucial for the overall decoder performance: a very bad estimate $\underline{\hat{x}}^{(t)}$ could yield a sequence $\underline{\hat{y}}(w)^{(t+1)}$ that is farther away from $\underline{x}$ than $\underline{\hat{y}}(w)$ was, leading to a performance loss.

3.2.1 Viterbi Voronoi decoder

For the Viterbi-based decoder, we propose the following procedure: using the APP computed by the BP decoder, we estimate $\underline{\hat{x}}^{(t)}$ and then the sequence $\underline{\hat{y}}(w)^{(t+1)}$ is obtained by projecting

$\underline{\hat{x}}^{(t)}$ onto $\mathcal{V}_w$. Thanks to linearity, one can equivalently project $\underline{\hat{x}}^{(t)} \oplus \underline{\hat{y}}(w)$ onto $\mathcal{V}_0$, yielding an output sequence $\underline{v}^*$ such that $\underline{\hat{y}}(w)^{(t+1)} = \underline{v}^* \oplus \underline{\hat{y}}(w)$. Thus, the new sequence $\underline{\hat{y}}(w)^{(t+1)}$ is given by:

$$\underline{\hat{y}}(w)^{(t+1)} = \underline{\hat{y}}(w) \oplus \arg\min_{\underline{v} \in \mathcal{V}_0} d_H \left(\underline{v}, \underline{\hat{x}}^{(t)} \oplus \underline{\hat{y}}(w) \right). \tag{3.1}$$

The remaining question for this decoder is what LLRs should one feed to the BP decoder. Since the projection step using the Viterbi algorithm is a hard-output sequence, we only have the LLRs associated to $\underline{\hat{y}}(w)$. One solution could be to assign to VN i

$$LLR_i = \begin{cases} 0 & \text{if } \hat{y}_i(w) \neq \hat{y}_i(w)^{(t+1)} \\ LLR(\hat{y}_i(w)) & \text{else.} \end{cases} \tag{3.2}$$

Note that setting a LLR to zero is equivalent to erasing this bit position. Doing so, we don't take into account at all the reliability of the estimate $\underline{\hat{x}}^{(t)}$ and this results in very poor performance. Based on our experiments, we propose the following heuristic LLR update rule:

$$LLR_i = \begin{cases} 0.9^j * LLR(\hat{y}_i(w)) & \text{if } \hat{y}_i(w) \neq \hat{y}_i(w)^{(t+1)} \\ LLR(\hat{y}_i(w)) & \text{else,} \end{cases} \tag{3.3}$$

where $j-1$ is the number of times a Voronoi projection has already been performed. Thus, the more projections (and decoder iterations) have been made, the closer $\underline{\hat{x}}^{(t)}$ is assumed to be to the source sequence, while in the first projections, the estimate may be far away from the source sequence. So we don't modify the LLRs too aggressively during the first projections, since the LDPC decoder would have difficulties to correct them, while in later iterations this should be no longer a problem. The factor 0.9 was determined experimentally to yield good results in our simulation setup; it may need to be changed for other setups.

Results obtained with a non-optimized rate $R_X = 0.1$ LDPC code of size $n = 1000$ and the rate $R_Y = 1/2$ convolutional code (5,7), with generator matrix $[1 + D^2, 1 + D + D^2]$ are shown in Figure 3.3. The sets of curves are indexed by the crossover probability p of the BSC relating X and Y. We can observe that the proposed method increases the rate of successful convergence. We perform $T = 30$ decoder iterations with the LLR associated with $\hat{Y}(w)$. In case of failure of the decoder, we perform up to 18 searches of $\hat{Y}(w)^{(T+lt+1)}$, and for each, we perform up to $t = 15$ decoder iterations. The results are given for 10000 samples.

This experimental setup was sufficient to show the feasibility of our method, but its overall performance is not overwhelming. To overcome this, one has to use optimized LDPC codes.

Since good low-rate LDPC codes are rather hard to come by, we decided to increase the rate of the encoder E_Y in order to be able to use rate-1/2 LDPC codes optimized for the BSC. Various optimized high-rate convolutional codes can be found in the literature, see for example [Tang and Lin, Jan. 2002], [Lin and Costello, 2004], [Lee, 1985].

For our second experiment, we chose an optimized rate-1/2 LDPC code with variable node degrees

$$\lambda(x) = 0.24426x + 0.25907x^2 + 0.01054x^3 + 0.05510x^4 + 0.01455x^7 + 0.01275x^9 + 0.40373x^{11}$$

and concentrated check node degrees, found in [Richardson and Urbanke, 2001a]; parity check matrices were constructed with the PEG algorithm.

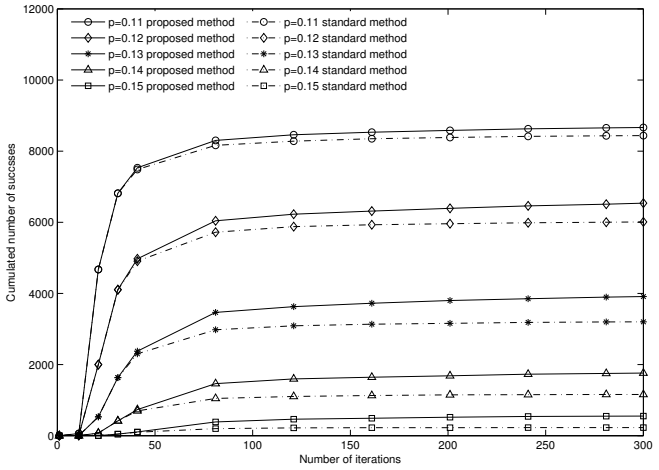


Figure 3.3: Comparison between the standard and the proposed method using a rate-0.1 LDPC code, the rate-1/2 convolutional code (5,7) and the Viterbi algorithm to perform the projection: Cumulated number of decoding successes as a function of the number of iterations.

For the quantizer, we used a rate-5/6 convolutional code found in [Tang and Lin, Jan. 2002] with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 \\ 2 & 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 3 \end{bmatrix}. \tag{3.4}$$

This generator matrix is given using the octal notation. This code uses 3 binary memory cells, meaning there are 8 trellis states. Increasing the encoder memory will improve the performance of the code in terms of distortion. At the same time, it will increase the number of trellis states and strongly expand the number of states in the Voronoi FSM, so we will only consider codes with 3 binary memory cells.

For this experimental setup, we set the code length to $n = 1200$.

We can see in Figure 3.4 that even with an optimized LDPC code and a high-rate convolutional code, the proposed method outperforms the standard one.

3.2.2 BCJR Voronoi decoder

The Viterbi Voronoi decoder has the conceptual advantage of delivering a sequence that is guaranteed to lie in $\mathcal{V}_w$ and thus to have been a possible quantizer input sequence. Nevertheless, the difficulty is to find an optimal LLR update rule. An obvious alternative is to use the soft-input, soft-output BCJR algorithm [Bahl et al., Mar. 1974] on the Voronoi FSM trellis.

For the soft version based on the BCJR algorithm, the procedure is a little bit different since we use soft values as inputs of the projection step, thus we can directly take into account the reliability of $\hat{\underline{x}}^{(t)}$. Moreover, in the first projections, the decided output sequence should be close to the all zero-sequence, in order not to modify the LLRs given to the BP decoder too aggressively. As inputs of the BCJR algorithm, we use scaled version of the extrinsic z_i computed by the BP algorithm: $z_i = \sum_{c_j \in \mathcal{N}(v_i)} m_{j,i}^{c \to v}$, where $m_{j,i}^{c \to v}$ denotes the message passed

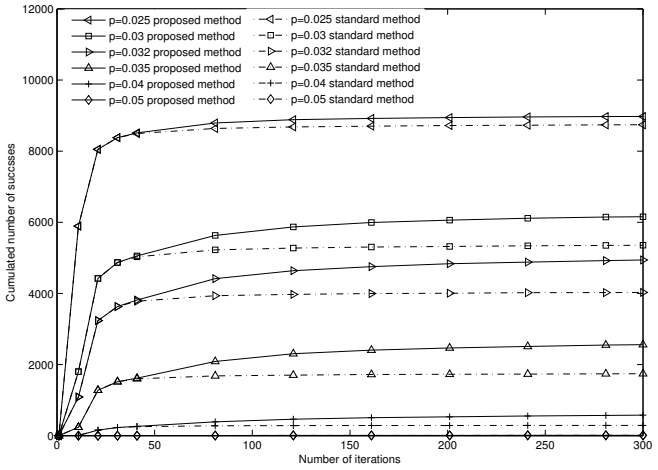


Figure 3.4: Comparison between the standard and the proposed method using a rate-0.5 optimized LDPC code, a rate-5/6 convolutional code and the Viterbi algorithm to perform the projection: Cumulated number of decoding successes as a function of the number of iterations.

from check node j to variable node i. Inputs of the BCJR algorithm are

$$LLR_i^{(\mathrm{BCJR})} = (1 - 2\hat{y}_i(w))(1 - 0.99^j)z_i, \tag{3.5}$$

where $j-1$ is the number of times a Voronoi projection has already been performed. This heuristic scaling is done to take into account the reliability of the estimate $\underline{\hat{x}}^{(t)}$, as previously. The sign flipping operation is done to perform on the Voronoi cell $\mathcal{V}_0$.

The factor 0.99 was again chosen based on the good performance obtained in our simulation setup, it may need to be changed for other setups.

The BCJR then computes the bit-wise APP values, i.e. the marginal posterior likelihood of bit y_i given the quantizer index w and a soft estimate of the source. From these we obtain an extrinsic soft-output LLR sequence $extr$.

Since the BCJR outputs an APP for Y and since $X = Y \oplus N$, with N a Bernoulli-p source, we can use the tanh-rule to compute the new "channel" LLR fed to the LDPC decoder for the next t iterations.

$$LLR_i = 2\tanh^{-1}\left(\tanh\left(\frac{extr_i}{2}\right)\tanh\left(\frac{\log\left(\frac{1-p}{p}\right)}{2}\right)\right). \tag{3.6}$$

For simulations with the BCJR, we used an optimized rate-1/2 LDPC code of length $n = 1200$ and first performed $T = 20$ decoding iterations with the LLR associated with $\hat{Y}^n(w)$. In case of failure of the decoder, we then perform up to 76 BCJR runs, after each of which we perform up to $t = 5$ LDPC decoding iterations. The results are given for 10000 samples. Results are given on Figure 3.5.

For completeness, we also ran simulations with a soft-input Viterbi decoder; the obtained results are shown in Figure 3.6 and compared to the standard and BCJR methods in Figure 3.7. The gain over the standard variant is only slightly above that for hard-input Viterbi, indicating the importance of proper soft outputs.

Since the decoding threshold is clearly seen to be shifted in Figure 3.7, we computed the theoretical thresholds (Shannon limits) for comparison purposes. An ideal rate-5/6 binary code

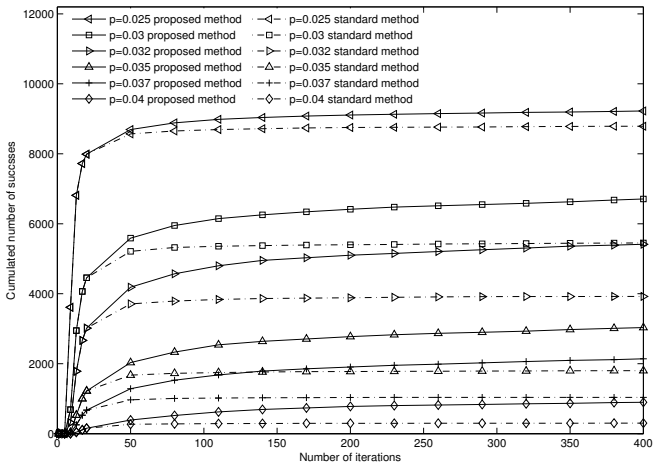


Figure 3.5: Comparison between the standard and the proposed method using a rate-0.5 optimized LDPC code, a rate-5/6 convolutional code and the BCJR algorithm to perform the projection: Cumulated number of decoding successes as a function of the number of iterations.

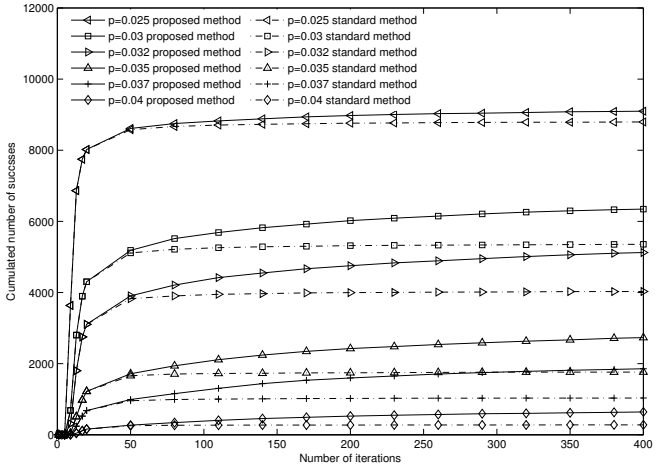


Figure 3.6: Comparison between the standard and the proposed method using a rate-0.5 optimized LDPC code, a rate-5/6 convolutional code and the the soft-input Viterbi algorithm to perform the projection: Cumulated number of decoding successes as a function of the number of iterations.

has theoretical distortion $D^*_{th} = 0.0246$, which yields $p^*_{th} = 0.0898$. The actual distortion obtained for our considered convolutional code is $D = 0.0373$, yielding $p_{th} = 0.0786$. These thresholds are still far from the ones observed in our simulations, which may partly be due to the moderate LDPC block length. Nevertheless, we can see that the proposed method provides clearly better performance than the standard one. A possible explanation of this gain is that the Voronoi decoder is able to compensate part of the quantizer suboptimality.

Beyond its superior performance (at the price of complexity), the BCJR Voronoi decoder has another major interest: it can be used to perform numerical density evolution of the overall

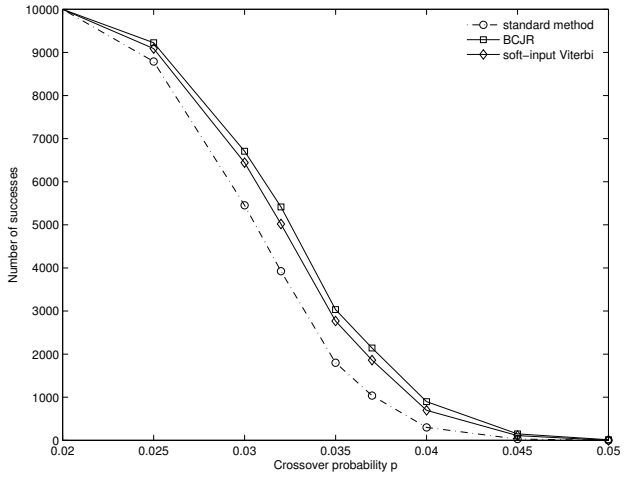


Figure 3.7: Comparison between the standard and the proposed methods using a rate-0.5 optimized LDPC code, a rate-5/6 convolutional code, 400 decoding iterations: Number of decoding successes (out of 10000 samples) as a function of source correlation parameter p.

decoder, along the lines of the approach presented in [Kavčić et al., 2003]. This allows to match the optimization of the LDPC code to the structure of the Voronoi cell.

3.3 Optimization of the LDPC degree distribution

As presented in Chapter 1, one way to optimize LDPC codes is to run a density evolution and a differential evolution.

The main difference between density evolution used in a standard channel coding situation and for our decoder is the following: in channel coding, one key assumption is that the all-zero codeword has been sent, thus the error probability tracked is defined as $P_e^{(l)} = \int_{-\infty}^{0} f_v^{(l+1)}(x)dx$. Unfortunately, this assumption doesn't hold for our setup, thus we must frequently change from/to the Voronoi cell $\mathcal{V}_0$ or $\mathcal{V}_w$. A major contribution was to show that using the BCJR Voronoi decoder, our improved decoder is still amenable to density evolution. Our approach follows the ideas proposed in [Kavčić et al., 2003], where the authors used a numerical density evolution using a BCJR algorithm over a binary intersymbol interference channel.

The following notations will be used:

- $f_v^{(l)}$: pdf of message from a VN to a CN at lth iteration;
- $f_c^{(l)}$: pdf of message from a CN to a VN at lth iteration;
- $f_o^{(l)}$: pdf of a priori LLR at the lth iteration;
- $f_e^{(l)}$: pdf of extrinsic given to the BCJR at lth iteration.

The computations of $f_v^{(+1)}$, $f_c^{(l+1)}$ and the initialization of $f_o^{(1)}$ are carried out as outlined in Chapter 1 for a BSC-ϵ.

The only step that differs from the standard density evolution is the one computing $f_o^{(l+1)}$: since there isn't a closed-form expression for this density, it will be computed numerically using Monte-Carlo techniques.

The Voronoi decoder transforms $f_e^{(l)}$ into $f_o^{(l+1)}$ as follows:

$$f_o^{(l)} = \begin{cases} f_o^{(l-1)}, & \text{if } l \neq kt \text{ (we don't perform a projection during this iteration)} \\ \epsilon_{trellis}(f_e^{(l)}, p), & \text{otherwise} \end{cases} \tag{3.7}$$

where $\epsilon_{trellis}$ is a symbolic notation for trellis evolution.

The average density of the extrinsic given to the BCJR is obtained by

$$f_e^{(l)} = \sum_{i=1}^{V_{max}} \tilde{\lambda}_i \left(\bigotimes_{k=1}^{i} f_c^{(l)} \right) \tag{3.8}$$

where $\tilde{\lambda}_i = \lambda_i / \left(i \int_0^1 \lambda(x) dx \right)$ is the fraction of variable nodes of degree i and $\bigotimes$ denotes convolution.

The density evolution associated to the proposed decoder is described in Algorithm 5. We first start by performing T iterations of standard LDPC density evolution, before performing the first projection step. After this first projection, t iterations of density evolution are performed before each additional projection. In total, *proj* projection are performed.

The operation in line 5 of Algorithm 5, corresponding to the numerical density evolution obtained via Monte Carlo simulation, is detailed in Algorithm 6.

Algorithm 5 Density evolution with BCJR Voronoi decoder

1: Initialization:
   $f_o^{(1)} = \epsilon\delta\big(x + \log\left(\frac{1-\epsilon}{\epsilon}\right)\big) + (1-\epsilon)\delta\big(x - \log\left(\frac{1-\epsilon}{\epsilon}\right)\big)$
   $P_e^{(1)} = 1$;
2: Density evolution: Phase 1 T BP iterations using $f_o^{(1)}$
3: **for** $j = 1 \rightarrow proj$ **do**
4: $\quad f_e^{(T+(j-1)t)} = \sum_{i=1}^{V_{max}} \tilde{\lambda}_i \left(\bigotimes_{k=1}^{i} f_c^{(T+(j-1)t)} \right)$
5: $\quad$ Trellis evolution: $f_o^{(T+(j-1)t)} = \epsilon_{trellis}(f_e^{(T+(j-1)t)}, p)$
6: $\quad$ Density evolution: Phase 2 t BP iterations using $f_o^{(T+(j-1)t)}$
7: **end for**

Algorithm 6 Trellis evolution $\epsilon_{trellis}(f_e^{(l)}, p)$

1: Draw an i.i.d. vector V of distribution $f_o^{(l)}$
2: $V \leftarrow V \times \tilde{X} \times \tilde{Y}$ where $\tilde{X} \in \{-1, 1\}^n = 1 - 2X$ and $\tilde{Y} = 1 - 2\hat{Y}(w)$
3: Account for the estimate reliability according to (3.5)
4: $V_1 \leftarrow$ BCJR-based Voronoi decoder
5: $V_1 \leftarrow V_1 \times \tilde{Y}$
6: $LLR_i = 2\tanh^{-1}\left(\tanh\left(\frac{V_{1_i}}{2}\right) \tanh\left(\frac{\log\left(\frac{1-p}{p}\right)}{2} \right) \right)$
7: $\widetilde{LLR} \leftarrow LLR \times \tilde{X}$
8: $f_o^{(l)} \leftarrow$ Density of $\widetilde{LLR}$

Since the Voronoi projection step has to be performed for an i.i.d. binary sequence to obtain proper averages (see [Kavčić et al., 2003] for more details), we first draw a random sequence X

and compute the corresponding quantization index. Then the BCJR Voronoi projection method is used to compute a realization of an extrinsic vector. In order to avoid trellis boundary effects, the block length n has to be rather large.

The following decoding thresholds were obtained using this numerical density evolution: $p^* = 0.0337$ for the standard method and $p^* = 0.0368$ for the proposed decoder, matching the performance observed on Figure 3.7.

The density evolution just outlined can then be used to optimize irregular LDPC degree distribution using differential evolution [Storn and Price, 1997]; see [Richardson and Urbanke, 2001a] for the application to LDPC channel codes. For more details about this method, we refer the reader to Chapter 1.

Using differential evolution and numerical density evolution, we found a rate-1/2 LDPC code ensemble, of decoding threshold $p^* = 0.06$, with variable degree distribution

$$\lambda(x) = 0.094167x^2 + 0.7275x^3 + 0.0125x^5 + 0.045x^6 + 0.00417x^{10} + 0.0317x^{14} + 0.0233x^{15} + 0.000833x^{16} + 0.04583x^{19} + 0.015x^{20}$$

and concentrated check-node degrees. This code has been optimized for the same quantizer as in the previous simulations.

Figure 3.8 compares the results obtained by the standard method and the improved decoder, using either the rate-1/2 code found in [Richardson and Urbanke, 2001a] (denoted 'without optimization') or the optimized rate-1/2 code (denoted 'with optimization'). For both LDPC codes, our method outperforms the standard one and the decoding threshold is shifted. Experiments with longer block lengths show that the threshold shift vanishes for the optimized codes, i.e. the advantage of knowing $\mathcal{V}_0$ is mainly exploited by the code optimization and so there is little gain left for the Voronoi decoder. Conversely, a non-optimized code will always benefit from the Voronoi decoder.

Recall that an ideal rate-5/6 quantizer has theoretical distortion $D^*_{th} = 0.0246$, which yields $p^*_{th} = 0.0898$. The actual distortion of our considered convolutional quantizer is $D = 0.0373$, yielding $p_{th} = 0.0786$. For now, the best rate-1/2 degree distributions that we found have a threshold of $p = 0.065$, which is still far from the theoretical limits. An explanation for this gap may lie in the heuristic scaling rule used before the projection, thus it is likely that density evolution won't be able to close this gap. To overcome this problem, a refined model of the processing chain $X - Y - \hat{Y}(W)$ must be found in order to use the Voronoi FSM to directly estimate X, without having to approximate the reliability of the projection step.

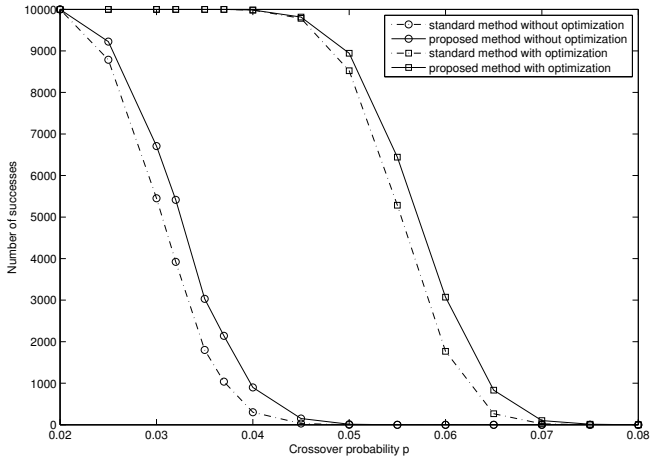


Figure 3.8: Comparison between the standard and the proposed method using a rate-0.5 LDPC code (either found in the literature or optimized for the overall decoder), a rate-5/6 convolutional code, 400 decoding iterations: Number of decoding successes as a function of the crossover probability p.

Summary and conclusion of Part I

The first part of this thesis was centered around an improved scheme for the binary source coding problem with coded side information, where we used a convolutional code to compress the side information and a LDPC decoder performing in Slepian-Wolf fashion.

In a standard setup, the side information is conveyed by a single quantization index, but many sequences are mapped onto the same index by the quantizer. The key point of our new decoding principle consists in projecting an intermediate estimate of the source sequence, given by the BP decoder, onto the Voronoi cell associated to the quantization index.

In order to do so, one needs a characterization of all Voronoi cells of the convolutional code used in the side information branch. Thanks to linearity, a characterization of the fundamental Hamming-space Voronoi cell $\mathcal{V}_0$ is sufficient. This was realized using a particular FSM. We first presented how to describe the Viterbi decoder of a convolutional code using a FSM and then how to build the graph of the FSM associated to the Voronoi cell $\mathcal{V}_0$, by studying the evolution of the Viterbi algorithm in terms of metric differences of the sequences mapped onto the all-zero codeword. Finally, we presented how to assign transition probabilities to this FSM in order to obtain an uniform probability distribution over all sequences in $\mathcal{V}_0$.

A way to possibly simplify this FSM is based on an observation made for perfect codes: Suppose that the minimum distance of the perfect code equals d_{min}, then the Voronoi cell $\mathcal{V}_0$ is the set of all sequence of weight up to $\frac{d_{min}-1}{2}$, which can be described by a counter. Hence, maybe an FSM with windowed counters may represent a relatively large subset of $\mathcal{V}_0$, that is sufficient for LDPC optimization. Having such a simplified FSM would of course reduce the complexity of the projection step onto the cell in our application, respectively make it possible at all. This characterization can be applied to all linear block code trellises.

Using the FSM-$\mathcal{V}_0$, we proposed a new decoding principle for the coded side information problem. By projecting an intermediate estimate of the source word onto $\mathcal{V}_0$, we can improve the side information given to the main decoder and thus accelerate the decoding of the source word.

Furthermore, we showed that this new decoder is amenable to density evolution and thus to LDPC code optimization. A first step was to adapt density evolution to this decoder by performing a numerical step via Monte Carlo simulations for the evolution of density during the projection step, for which there is no closed-form expression. Finally, differential evolution is used to optimize an irregular LDPC degree profile for the convolutional code of the side information branch.

Results show a significant gain in term of number of decoding successes vs. the number of decoding iterations, when optimized codes and the improved method are used. For large block sizes, the knowledge of $\mathcal{V}_0$ is completely resorbed in the code optimization, that is the (complex) Voronoi decoder step is only needed off-line.

Nevertheless, there remains a gap between theoretical and experimental thresholds to close. A possible explanation for this gap lies in the heuristic update rule modeling the reliability of the source estimate. To overcome this problem, one could investigate a better model for the chain $X - Y - \hat{Y}(w)$ to directly estimate X with the Voronoi FSM. Another major improvement could consist in using non i.i.d. LDPC codes: since the side information is compressed using a convolutional code, the output is in general no longer i.i.d.

Part II

Relay channels

Overview

The simplest model of cooperative communication is the three-node *relay channel*, where one source node wishes to send data to a destination with the help of one relay. This model has been widely studied, but its capacity remains unknown in general.

The relay channel can also been seen as the combination of a Multiple-Access Channel (MAC), formed by the links relay-destination and source-destination, and a broadcast channel, formed by the links source-relay and source-destination. For the broadcast channel, a source sends its message to many receivers that recover this data, thus its capacity is always limited by the noisiest receiver, whereas for the MAC, multiple sources send their data to one single destination, that recovers all messages, thus for two users, the maximal sum rate of $C\left(\frac{P_1+P_2}{N}\right)$ can be achieved.

Various relaying schemes have been proposed in order to perform close to an upper-bound of the capacity. The difference between these relaying schemes lies in the operation performed by the relay: it can for example decode the message sent by the source (in the Decode-and-Forward protocol), or amplify its received signal (in the Amplify-and-Forward protocol), or quantize its received signal (in the Compress-and-Forward protocol). First, all of these lower bounds were proved using AWGN random coding, but in the past years, lattices have been shown to achieve the same rates. Moreover, for some extensions of the relay channel, such as the two-way relay channel, where two users wish to exchange their data with the help of one relay, relaying schemes exploiting the algebraic structure of lattices such as Compute-and-Forward, in which the relay directly decodes the sum of the two messages sent, have been proposed.

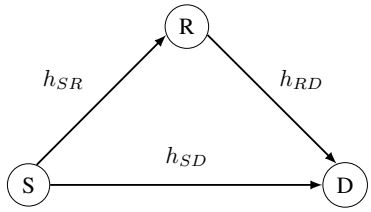
In this part of the thesis, we focus on the study of various extensions of the relay channel. For all channel models, the aim is to derive upper and lower bounds on the capacity. The proofs are either based on AWGN coding or lattice coding. All the considered models are given on Figure 3.9. The links between nodes are labeled with the respective channel gain. The noise(s) at the receiver(s) are assumed to be AWGN.

We first recall some results and the associated proofs for the full-duplex Gaussian relay channel (given on Figure 3.9a).

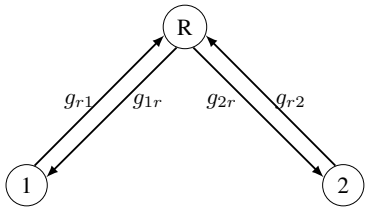
The first extension considered is a generalization of the relay channel considering correlated noises at the relay and destination.

The second extension considers again three nodes: two source/destination nodes and one relay. The two users wish to exchange their data with the help of a relay. Three models are studied: the two-way relay channel without direct links between the two users (given on Figure 3.9b), where one needs a multi-hop protocol, the two-way relay channel with direct links between the two users (given on Figure 3.9c) and again a generalization of the two-way relay channel with direct links between the two users and correlated noises at the relay and the users.

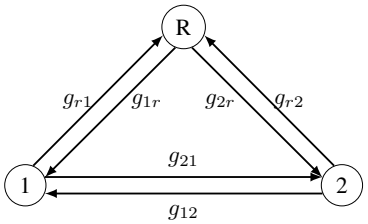
The last extension considers more than three nodes. In the multiway relay channel (given on Figure 3.9d), multiple clusters of users wish to exchange messages locally within each cluster with the help of a single relay. Users within a cluster are fully connected. First, this channel is studied assuming time-sharing among the clusters and then the study is performed on the relaxed model (i.e. without time-sharing).



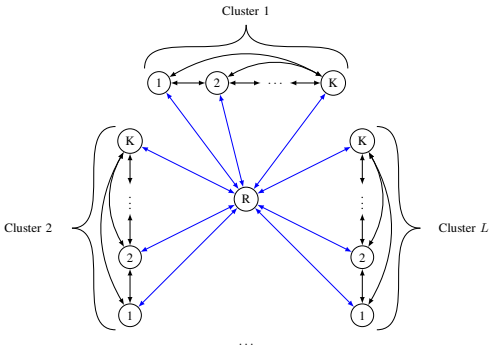
(a) Relay channel



(b) Two-way relay channel without direct links



(c) Two-way relay channel with direct links



(d) Multiway relay channel

Figure 3.9: Relay channel models

Chapter 4

The Gaussian relay channel

Part of this work was presented in [Savard and Weidmann, 2015b].

Contents

The relay channel model, introduced by [van der Meulen, 1971] and studied by [Cover and El Gamal, Sept. 1979] is one of the fundamental building blocks in wireless communications. A source wishes to send its message to a destination with the help of a relay: this occurs for example for communications between two base stations through a terrestrial link and a satellite, or in a network with an intermediate node acting as a relay. The relay channel is depicted on Figure 4.1.

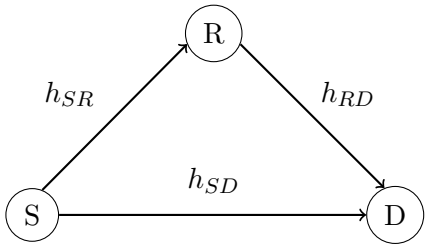


Figure 4.1: Relay channel

Information may thus flow along the direct or the relayed link. The capacity of this channel is unknown in general, but a cut-set (upper) bound can be established. The key question in approaching the capacity is how the two links should cooperate. Two extreme cases can be considered:

- direct transmission, where the relay is not used in the communication.
- Decode-and-Forward (DF), where the relay decodes the message and coherently cooperates with the source to communicate it to the destination. This includes pure multi-hop schemes in which the direct link is not exploited.

Other schemes have also been proposed, such as

- Partial-Decode-and-Forward, where the relay decodes a part of the message and the remainder of the message is decoded only at the destination.
- Compress-and-Forward(CF), where instead of recovering the message, the relay compresses it using Wyner-Ziv coding techniques, with the sequence received at the destination acting as side information.
- Amplify-and-Forward (AF), where the relay only scales its received message up to its power constraint before transmitting it.

Throughout this chapter, we focus only on full-duplex nodes, which means that a node can receive and transmit at the same time.

We study two Gaussian relay channel models: the first one is the standard full-duplex Gaussian relay channel and the second one a more general full-duplex Gaussian relay channel with correlated noises. The aim of the first section is to present the Gaussian relay channel, as well as standard protocols and the way to prove achievable rates.

For the standard relay channel, we briefly summarize results found e.g. in [El Gamal and Kim, 2011], [Cover and El Gamal, Sept. 1979] or [Kramer et al., Sept. 2005]. Even if the proofs are quite usual, we recall them since they present the main techniques used for various protocols proposed in this thesis.

4.1 Standard full-duplex Gaussian relay channel

In the Gaussian case, the source sends X_1 of power P_1 and the relay X_R of power P_R. The received signals are given by:

$$\text{At the relay: } Y_R = h_{SR}X_1 + Z_R,$$
$$\text{At the destination: } Y_2 = h_{SD}X_1 + h_{RD}X_R + Z_D,$$

where Z_D and Z_R are independent additive white Gaussian noises of variance N_D and N_R, respectively.

Even if the relay channel is the main building block of cooperative communications, its capacity remains unknown. Thus, we start by introducing an upper bound on the capacity (using a cut-set bound) and we then present protocols that can perform close to this cut-set bound.

4.1.1 Cut-set bound [El Gamal and Kim, 2011]

A multicast network is modeled via a directed acyclic graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{C})$, where $\mathcal{N} = [1 : N]$ is the set of nodes, $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the set of edges and $\mathcal{C} = \{C_{i,j}, i, j \in \mathcal{E}\}$ is the set of edge weights. This situation is depicted on Figure 4.2. Each node represents a transmitter and/or a destination and each edge (i, j) represents a noiseless communication link from node i to node

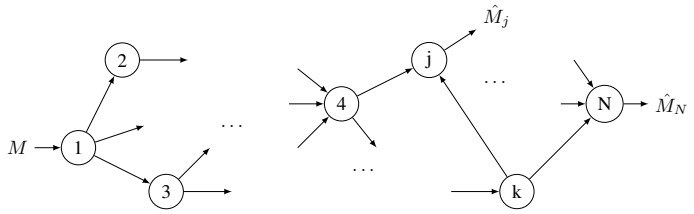


Figure 4.2: Example of a graphical multicast network

j of capacity $C_{i,j}$. The source node 1 wishes to communicate a message $M \in [1 : 2^{nR}]$ to a set of destinations $\mathcal{D} \subset \mathcal{N}$. Each node $k \in [2, N]$ can act as a relay.

The average error probability is defined as $P_e^{(n)} = P(\hat{M}_j \neq M \text{ for some } j \in \mathcal{D})$. A rate R is said to be achievable if there exists a sequence of $(2^{nR}, n)$ codes such that $\lim_{n\to\infty} P_e^{(n)} = 0$. The capacity is defined as the supremum of the set of achievable rates.

An upper bound on the capacity is obtained considering cut-set arguments.
For a destination $j \in \mathcal{D}$, we can define a cut $(\mathcal{S}, \mathcal{S}^c)$ as a partition of nodes such that node 1 (i.e. the source node) is in set $\mathcal{S}$ and the destination node j is in set $\mathcal{S}^c$. The capacity of this cut is given by $C(\mathcal{S}) = \sum_{\substack{(k,j)\in\mathcal{E} \\ k\in\mathcal{S}, l\in\mathcal{S}^c}} C_{kl}$.

The capacity of the network cannot be larger than the smallest cut capacity $C(\mathcal{S})$ for every destination node $j \in \mathcal{D}$, which is formalized in the following theorem.

Theorem 4.1.1. *The capacity of the multicast network $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{C})$ with destination set $\mathcal{D}$ is upper bounded as $C \leq \min_{j\in\mathcal{D}} \min_{\substack{\mathcal{S}\subset\mathcal{N} \\ 1\in\mathcal{S}, j\in\mathcal{S}^c}} C(\mathcal{S})$.*

We can also consider a graphical unicast model, where there is only one destination node. Without loss of generality, assume that $\mathcal{D} = N$. A unicast network with $N = 5$ nodes is given on Figure 4.3. We also give an example of a cut on the same figure.

In this case Theorem 4.1.1 reduces to the max-flow min-cut theorem.

Theorem 4.1.2. *The capacity of the unicast network $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{C})$ with destination node N is $C = \min_{\substack{\mathcal{S}\subset\mathcal{N} \\ 1\in\mathcal{S}, N\in\mathcal{S}^c}} C(\mathcal{S})$.*

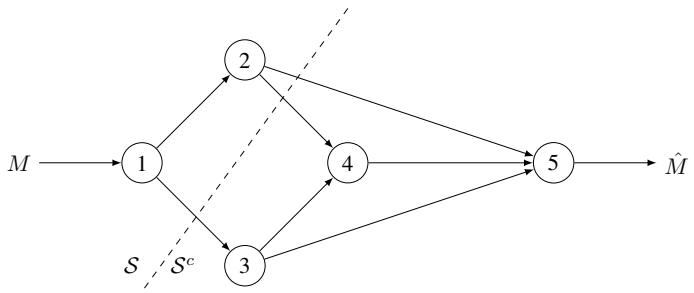


Figure 4.3: Example of a graphical unicast network with one cut

Proposition 4.1.1. *([Cover and El Gamal, Sept. 1979]) For the Gaussian relay channel, the*

Cut-Set Bound (CSB) is given by

$$R_{CSB} = \max_{\rho\in[0,1]} \min\left\{ C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\rho\sqrt{P_1 P_R}}{N_D}\right), C\left(\frac{P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R)}{N_R N_D}\right)\right\},$$

where $C(x) = \frac{1}{2}\log_2(1+x)$ *and* $\overline{x} = 1 - x$.

Proof. Using Theorem 4.1.2, the capacity of the Gaussian relay channel is upper bounded by

$$C \le \max_{p(x_1,x_R)} \min\{I(X_1, X_R; Y_D); I(X_1; Y_R, Y_D|X_R)\}$$

We write the first mutual information as

$$\begin{aligned} I(X_1, X_R; Y_D) &= H(Y_D) - H(Y_D|X_1, X_R) \\ &= H(Y_D) - \frac{1}{2}\log_2(2\pi e N_D) \\ &\le \frac{1}{2}\log_2\left(\frac{\mathbb{E}[Y_D^2]}{N_D}\right) \\ &\le C\left(\frac{h_{SD}^2\mathbb{E}[X_1^2] + h_{RD}^2\mathbb{E}[X_D^2] + 2h_{SD}h_{RD}\mathbb{E}[X_1 X_R]}{N_D}\right) \\ &\le C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\rho\sqrt{P_1 P_R}}{N_D}\right), \end{aligned}$$

where $\rho = \dfrac{\mathbb{E}[X_1 X_R]}{\sqrt{P_1 P_R}}$ is the correlation coefficient between X_1 and X_R.

The second mutual information can be written as

$$\begin{aligned} I(X_1; Y_R, Y_D|X_R) &= H(Y_R, Y_D|X_R) - H(Y_R, Y_D|X_1, X_R) \\ &= H(Y_R, Y_D|X_R) - H(Z_R, Z_D) \\ &= H(Y_D|X_R) + H(Y_R|X_R, Y_D) - \frac{1}{2}\log_2\left(4\pi^2 e^2 N_D N_R\right) \\ &\le \frac{1}{2}\log_2(2\pi e\mathbb{E}[\mathrm{Var}(Y_D|X_R)]) + \frac{1}{2}\log_2(2\pi e\mathbb{E}[\mathrm{Var}(Y_R|X_R, Y_D)]) \\ &\quad - \frac{1}{2}\log_2\left(4\pi^2 e^2 N_D N_R\right) \\ &\overset{(a)}{\le} \frac{1}{2}\log_2\left(2\pi e(N_D + h_{SD}^2 P_1\overline{\rho^2})\right) + \frac{1}{2}\log_2\left(2\pi e\frac{P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R) + N_R N_D}{h_{SD}^2 P_1\overline{\rho^2} + N_D}\right) \\ &\quad - \frac{1}{2}\log_2\left(4\pi^2 e^2 N_D N_R\right) \\ &\le C\left(\frac{P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R)}{N_R N_D}\right), \end{aligned}$$

where (a) follows from the fact that the conditional mean squared error of the linear MMSE estimate of Y given X upper bounds the expected variance $\mathbb{E}[\mathrm{Var}(Y|X)]$.

Using the linear MMSE estimate of Y_D given X_R, we obtain

$$\mathbb{E}[\mathrm{Var}(Y_D|X_R)] \le N_D + h_{SD}^2 P_1(1-\rho^2).$$

Using the linear MMSE estimate of Y_R given X_R and Y_D, we obtain

$$\mathbb{E}[\mathrm{Var}(Y_R|X_R,Y_D)] \leq \frac{P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R) + N_R N_D}{h_{SD}^2 P_1 \overline{\rho^2} + N_D}.$$

□

The remainder of the section presents various standard protocols, part of which can perform close to the cut-set bound.

4.1.2 Lattice-based Compress-and-Forward

In the Compress-and-Forward (CF) scheme, the relay helps the communication by sending a compressed description (a bin index) of its received message to the destination. Because this description is correlated with the source message, Wyner-Ziv coding can be used to reduce the rate needed at the relay. [Song and Devroye, Aug. 2013] showed that lattices can achieve the CF rate.

Lattices are a powerful tool that achieves capacity on the AWGN channel. It has first been showed that capacity-achieving codebooks can be obtained by intersecting a lattice with a 'thin' spherical shell [de Buda, Aug. 1989]. Later, [Urbanke and Rimoldi, Jan. 1998] showed that the intersection of a lattice with a spherical region can produce capacity-achieving codebooks. [Poltyrev, Mar. 1994] proved that lattice decoding on the AWGN channel, where the destination bins the received signal according to the Voronoi regions, is asymptotically efficient. [Loeliger, Nov. 1997] showed the existence of lattices that achieve $\frac{1}{2}\log_2(\mathrm{SNR})$ over the AWGN channel with lattice decoding and finally [Erez and Zamir, Oct. 2004] proved that using a modulo-lattice operation and random dither, lattice codebooks can achieve the AWGN capacity under lattice decoding.

Definition 4.1.1. *A lattice $\Lambda \subset \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$ closed under addition. In other words, $\forall \lambda_1, \lambda_2 \in \Lambda, \lambda_1 + \lambda_2 \in \Lambda, \lambda_1 - \lambda_2 \in \Lambda$.*

Equivalently a lattice Λ is the set of all integer combinations of a set of basis vectors. Regrouping this basis into a matrix yields the generator matrix $G \in \mathbb{R}^n$ of the lattice: A n-dimensional lattice Λ is defined as

$$\Lambda = \{GX : X \in \mathbb{Z}^n\}.$$

An example of a lattice is given on Figure 4.4.

Definition 4.1.2. *The lattice quantizer Q_Λ maps any point $x \in \mathbb{R}^n$ to the closest lattice point:*

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

Definition 4.1.3. *The lattice Λ partitions $\mathbb{R}^n$ into the union of Voronoi regions*

$$\mathcal{V}_\lambda = \{x \in \mathbb{R}^n | Q_\Lambda(x) = \lambda\}.$$

Definition 4.1.4. *The fundamental Voronoi region $\mathcal{V}_0$ of Λ is the set of points that are closer to the origin than to any other lattice point:*

$$\mathcal{V}_0 = \{x \in \mathbb{R}^n | Q_\Lambda(x) = 0\}, \text{ which is of volume } V = Vol(\mathcal{V}_0).$$

The fundamental Voronoi region of the hexagonal lattice is given on Figure 4.4.

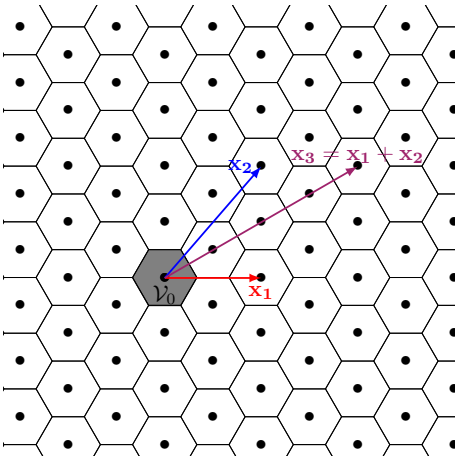


Figure 4.4: Example of a two-dimensional lattice

Definition 4.1.5. *The modulo Λ operation yields the quantization error:*

$$[x] \bmod \Lambda = x - Q_\Lambda(x), \text{ which is ensured to lie in } \mathcal{V}_0.$$

Definition 4.1.6. *The second moment per dimension $\sigma^2(\Lambda)$ defines the average power of the lattice Λ:*

$$\sigma^2(\Lambda) = \frac{1}{nV}\int_{\mathcal{V}_0} ||x||^2 dx, \text{ where } V = \int_{\mathcal{V}_0} dx \text{ is the volume of } \mathcal{V}_0.$$

Definition 4.1.7. *The normalized second moment of a lattice Λ of dimension n is defined as:*

$$G(\Lambda) = \frac{\sigma^2(\Lambda)}{V^{2/n}}.$$

It measures the efficiency of Λ as a shaping region: the normalized second moment of a sphere in $\mathbb{R}^n$ is $1/2\pi e$ and the more $\mathcal{V}_0$ resembles a sphere, the closer to $1/2\pi e$ $G(\Lambda)$ will be.

Definition 4.1.8. *The covering radius r_{cov} is the radius of the smallest sphere that covers $\mathcal{V}_0$:*

$$r_{cov} = \inf_{r>0}\{\mathcal{V}_0 \subset r\mathcal{B}^n\}, \text{ where } \mathcal{B}^n \text{ is the unit ball in } \mathbb{R}^n.$$

Definition 4.1.9. *The effective radius r_{eff} is the radius of a sphere with the same volume as $\mathcal{V}_0$:*

$$r_{eff} = \left(\frac{V}{Vol(\mathcal{B}^n)}\right)^{1/n}.$$

Good lattices for proving theoretic results need to satisfy some properties:

Definition 4.1.10. *A sequence of n-dimensional lattice $\Lambda^{(n)}$ is said Rogers-good [Rogers, 1959] if $\lim_{n\to\infty} \frac{r_{cov}^{(n)}}{r_{eff}^{(n)}} = 1$, that is if the covering radius approaches the effective radius.*

Definition 4.1.11. *A sequence of n-dimensional lattice $\Lambda^{(n)}$ is said Poltyrev-good [Poltyrev, Mar. 1994] (good for AWGN coding) if, for $Z \sim \mathcal{N}(0, \sigma^2 I)$, a n-dimensional vector, $Pr(Z \notin \mathcal{V}) \leq e^{-n(E_p(\mu)-o_n(1))}$, where $E_p(\mu)$ is the Poltyrev exponent and μ is the volume-to-noise*

ratio defined as $\mu = \frac{V^{2/n}}{2\pi e\sigma^2} + o_n(1)$.

The Poltyrev exponent is defined as $E_p(\mu) = \begin{cases} \frac{1}{2}[(\mu-1) - \log(\mu)] & \text{if } 1 < \mu \leq 2 \\ \frac{1}{2}\log\frac{e\mu}{4} & \text{if } 2 \leq \mu \leq 4 \\ \frac{\mu}{8} & \text{if } \mu \geq 4. \end{cases}$

Definition 4.1.12. *A sequence of n-dimensional lattice* $\Lambda^{(n)}$ *is said to be good for mean-square error quantization if*

$$\lim_{n\to\infty} G(\Lambda^{(n)}) = \frac{1}{2\pi e}.$$

It can be shown that if a lattice is Rogers-good, then it is also good for mean-squared error quantization [Zamir and Feder, Jul. 1996].

There exist different methods to build good lattices, such as constructions A, D, etc. [Conway et al., 1999], but they will not be presented in this thesis.

Good lattice codebooks are obtained with the help of two nested lattices Λ_c and Λ_f, such that $\Lambda_c \subseteq \Lambda_f$, with fundamental Voronoi region $\mathcal{V}_c$ of volume V_c and $\mathcal{V}_f$ of volume V_f respectively. An example of nested lattices is given on Figure 4.5.

Λ_c is called the *coarse lattice* (or *shapping lattice*) and Λ_f the *fine lattice* (or *coding lattice*). These lattices are chosen such that Λ_f is Poltyrev-good and Λ_c is both Rogers- and Poltyrev-good. The second moment per dimension of the coarse lattice is chosen to insure a power constraint. The rate of this codebook is $R = \frac{1}{n}\log_2\frac{V_c}{V_f}$.

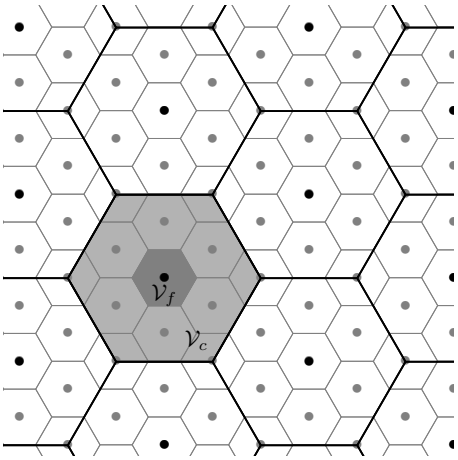


Figure 4.5: Example of nested lattices

We now present lattice coding and decoding for the AWGN channel, for which it has been shown that lattices are capacity-achieving [Erez and Zamir, Oct. 2004].

This channel is described by $Y = X + Z$, where $X \in \mathbb{R}^n$ is the transmitted signal, $Y \in \mathbb{R}^n$ the received one and $Z \in \mathbb{R}^n$ is additive Gaussian noise of variance N. We assume that X is limited to power P: $\frac{1}{n}||X||^2 \leq P$.

The codebook is obtained in the following way: Choose Λ_c and Λ_f as previously and such that $\sigma^2(\Lambda_c) = P$. The codebook is given by $\mathcal{C} = \Lambda_f \cap \mathcal{V}_c$.

To transmit the codeword $c \in \mathcal{C}$, the source sends $X = [c+U] \bmod \Lambda_c$, where U is a random dither uniformly distributed over $\mathcal{V}_c$ and is known at the source and the destination.

Lemma 4.1.1. *(Crypto lemma [Erez and Zamir, Oct. 2004]) For any random variable X distributed over the fundamental region $\mathcal{V}$ and statistically independent of U, which is uniformly distributed over $\mathcal{V}$, $[X+U] \bmod \Lambda$ is independent of X and uniformly distributed over $\mathcal{V}$.*

The destination receives $Y = X + Z$. To decode c, the destination scales its received signal by the MMSE coefficient $\beta = \frac{P}{P+N}$, subtracts the dither and takes the result modulo Λ_c:

$$\tilde{Y} = [c + \beta Z + (\beta - 1)X] \bmod \Lambda_c.$$

The receiver estimates c by quantization: $\hat{c} = Q_{\Lambda_f}(\tilde{Y})$. Erez and Zamir, [Erez and Zamir, Oct. 2004], showed that averaging over the dither, perfect decoding is possible if $R \leq C\left(\frac{P}{N}\right)$.

Proposition 4.1.2. *([Song and Devroye, Aug. 2013]) For the Gaussian relay channel, CF using lattices achieves the rate*

$$R_{CF} = C\left(\frac{P_1}{N_D}\frac{h_{SD}^2(N_R+D)+h_{SR}^2 N_D}{N_R+D}\right), \text{where } D = \frac{(h_{SR}^2 N_D + h_{SD}^2 N_R)P_1 + N_D N_R}{h_{RD}^2 P_R}.$$

Proof. The detailed proof has been proposed by [Song and Devroye, Aug. 2013].

The encoding and decoding procedures are based on block Markov coding.

- Encoding:
  The codebook for the sender is given by $\mathcal{C}_1 = \Lambda_{c1} \cap \mathcal{V}_1$, where $\Lambda_1 \subset \Lambda_{c1}$ and Λ_1 is both Rogers- and Poltyrev-good and Λ_{c1} is Poltyrev-good. To ensure the power constraints, we choose $\sigma^2(\Lambda_1) = P_1$ and Λ_{c1} such that $|\mathcal{C}_1| = R_{CF}$.

  During block b, the source sends $c_1(b) \in \mathcal{C}_1$ as

  $$X_1(b) = [c_1(b) + u_1(b)] \bmod \Lambda_1,$$

  where u_1 is a dither uniformly distributed over $\mathcal{V}_1$.

  The quantization codebook at the relay is given by $\mathcal{C}_q = \{\Lambda_{cQ} \cap \mathcal{V}_Q\}$, where $\Lambda_Q \subseteq \Lambda_{cQ}$ and Λ_{cQ} is Rogers-good and Λ_Q is Poltyrev-good. We choose the quantizer distortion $\sigma^2(\Lambda_{cQ}) = D$ and

  $$\sigma^2(\Lambda_Q) = h_{SR}^2 P_1 + N_R + D - \frac{(h_{SD}h_{SR}P_1)^2}{h_{SD}^2 P_1 + N_D}. \tag{4.1}$$

  Thus, the quantization rate is $R_q = \frac{1}{2}\log_2\left(\frac{\sigma^2(\Lambda_Q)}{D}\right)$.

  The codebook for the relay is given by $\mathcal{C}_R = \{\Lambda_{cR} \cap \mathcal{V}_R\}$, where $\Lambda_R \subseteq \Lambda_{cR}$ and Λ_R is both Rogers- and Poltyrev-good and Λ_{cR} is Poltyrev-good. To ensure the power constraint, we choose $\sigma^2(\Lambda_R) = P_R$. Each compression index $i \in \mathcal{C}_q$ is mapped to one codeword $c_R \in \mathcal{C}_R$, that is Λ_R is chosen s.t. $|\mathcal{C}_R| = R_q$.

  During block b, the relay sends

  $$X_R(b) = [c_R(I(b-1)) + u_R(b)] \bmod \Lambda_R,$$

  where u_R is a dither uniformly distributed over $\mathcal{V}_R$.

- Decoding:
  During block b, the relay receives $Y_R(b) = h_{SR}X_1(b) + Z_R(b)$ and quantizes it to

$$\begin{aligned} I(b) &= [Q_{cQ}(h_{SR}X_1(b) + Z_R(b) + u_{cQ}(b))] \bmod \Lambda_Q \\ &= [h_{SR}X_1(b) + Z_R(b) + u_{cQ}(b) - E_{cQ}(b)] \bmod \Lambda_Q, \end{aligned}$$

  where E_{cQ} is the quantization error and u_{cQ} is a dither uniformly distributed over $\mathcal{V}_{cQ}$.
  During block b, the destination receives

$$Y_D(b) = h_{SD}X_1(b) + h_{RD}[c_R(I(b-1)) + u_R(b)] \bmod \Lambda_R + Z_D(b).$$

  It starts by decoding the quantization index, considering the sender signal as noise, which is possible if

$$R_q \leq C\left(\frac{h_{RD}^2 P_R}{h_{SD}^2 P_1 + N_D}\right).$$

  Then, it forms $\widetilde{Y}_D(b) = h_{SD}X_1(b) + Z_D(b)$, by substracting the relay signal.
  The decoding of $X_1(b-1)$ is performed using Wyner-Ziv techniques. During the previous block, the destination formed $\widetilde{Y}_D(b-1)$ which is used in block b as side information to estimate $\hat{Y}_R(b-1)$, a noisy version of the signal received at the relay:

$$\begin{aligned} \hat{Y}_R(b-1) &= \left[I(b-1) - u_{cQ}(b-1) - \beta\widetilde{Y}_D(b-1)\right] \bmod \Lambda_Q + \beta\widetilde{Y}_D(b-1) \\ &= \left[h_{SR}X_1(b-1) + Z_R(b-1) - E_{cQ}(b-1) - \beta\widetilde{Y}_D(b-1)\right] \bmod \Lambda_Q + \beta\widetilde{Y}_D(b-1) \\ &= \left[(h_{SR} - \beta h_{SD})X_1(b-1) + Z_R(b-1) - E_{cQ}(b-1) - \beta Z_D(b-1)\right] \bmod \Lambda_Q \\ &\quad + \beta\widetilde{Y}_D(b-1) \\ &= h_{SR}X_1(b-1) + Z_R(b-1) - E_{cQ}(b-1). \end{aligned}$$

  The last equality is valid under perfect decoding, requiring

$$\sigma^2(\Lambda) \geq (h_{SR} - \beta h_{SD})^2 P_1 + N_R + D + \beta^2 N_D.$$

  Since we estimate $h_{SR}X_1(b-1) + Z_R(b-1)$ from $h_{SD}X_1(b-1) + Z_D(b-1)$, the linear MMSE orthogonality principle requires that β is chosen as $\beta = \dfrac{h_{SD}h_{SR}P_1}{h_{SD}^2 P_1 + N_D}$.
  Thus,

$$\sigma^2(\Lambda) = h_{SR}^2 P_1 + N_R + D - \frac{(h_{SD}h_{SR}P_1)^2}{h_{SD}^2 P_1 + N_D}.$$

  Combining this with the quantization rate constraint, the distortion of the quantizer (in other words $\sigma^2(\Lambda_{cQ})$) becomes

$$D = \frac{(h_{SR}^2 N_D + h_{SD}^2 N_R)P_1 + N_D N_R}{h_{RD}^2 P_R}. \tag{4.2}$$

In order to recover $X_1(b-1)$, the receiver coherently uses two noisy observations of $X_1(b-1)$ ($\hat{Y}_R(b-1)$ and $\widetilde{Y}_D(b-1)$) as

$$\left(\frac{h_{SR}\sqrt{P_1}}{N_R+D}\right)\hat{Y}_R(b-1)+\left(\frac{h_{SD}\sqrt{P_1}}{N_D}\right)\widetilde{Y}_D(b-1) = X_1(b-1)\left(\frac{\left(h_{SD}^2(N_R+D)+h_{SR}^2N_D\right)\sqrt{P_1}}{N_D(N_R+D)}\right)$$
$$+ Z_D(b-1)\frac{h_{SD}\sqrt{P_1}}{N_D} + \left(Z_R(b-1) - E_{cQ}(b-1)\right)\frac{h_{SR}\sqrt{P_1}}{N_R+D}.$$

Thus, decoding succeeds if

$$R_{CF} \leq C\left(\frac{P_1}{N_D}\frac{h_{SD}^2(N_R+D)+h_{SR}^2N_D}{N_R+D}\right).$$

□

4.1.3 Decode-and-Forward

In Decode-and-Forward (DF), the relay plays a crucial role: it decodes the message and sends it coherently with the source to the destination. In this subsection, we show that lattices can achieve the DF rate [Nokleby and Aazhang, 2011]. The key to this approach relies in a specific lattice construction. Each message is the sum of two lattice points: one coarse lattice point, that the destination decodes alone, and one fine lattice point, that is decoded by both the relay and the destination. In this scheme, the relay decodes the fine lattice point and sends it to the destination coherently with the source. The destination starts by decoding this part of the message and then recovers the entire message.

Proposition 4.1.3. *([Nokleby and Aazhang, 2011]) For the Gaussian relay channel, DF using lattices achieves the following rate:*

$$R_{DF} = \max_{\alpha\in[0,1]} \min\left\{C\left(\frac{h_{SR}^2\bar{\alpha}P_1}{N_R}\right), C\left(\frac{h_{SD}^2P_1+h_{RD}^2P_R+2h_{SD}h_{RD}\sqrt{\alpha P_1P_R}}{N_D}\right)\right\}$$

α allows to trade off power at the source node between repeating the message from the previous block and sending a new message.

Proof. The detailed proof has been proposed by [Nokleby and Aazhang, 2011]. The encoding and decoding procedure is based on block Markov coding.

- Encoding:
  For the source, we use a doubly nested lattice coding scheme as proposed in [Nokleby and Aazhang, 2011] with $\Lambda_s \subseteq \Lambda_m \subseteq \Lambda_{c1}$, where Λ_s and Λ_m are both Rogers- and Poltyrev-good and Λ_{c1} is Poltyrev-good. An example of doubly nested lattices is depicted on Figure 4.6. As in the standard nested lattice coding scheme, Λ_s is the shaping lattice that ensures the power constraint and Λ_{c1} is the coding lattice. Λ_m is a meso lattice that groups codewords into clusters. Using these three lattices, we build the following three codebooks:

$$\begin{aligned}\mathcal{C}_1 &= \Lambda_{c1}\cap\mathcal{V}_s \text{ of rate } R_1\\ \mathcal{C}_{10} &= \Lambda_{c1}\cap\mathcal{V}_m \text{ of rate } R_{10}\\ \mathcal{C}_{11} &= \Lambda_m\cap\mathcal{V}_s \text{ of rate } R_{11}\\ (R_1 &= R_{10}+R_{11}).\end{aligned}$$

We set $\sigma^2(\Lambda_s) = 1$.

A codeword $c_1 \in \mathcal{C}_1$ can be written as $c_1 = [c_{10} + c_{11}] \bmod \Lambda_s$, where

$$c_{10} = c_1 \bmod \Lambda_m \in \mathcal{C}_{10} \text{ and}$$
$$c_{11} = [c_1 - c_{10}] \bmod \Lambda_s \in \mathcal{C}_{11}.$$

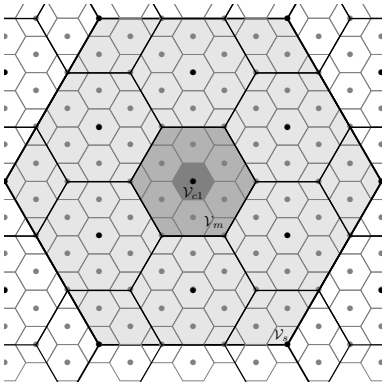


Figure 4.6: Doubly-nested lattices used for the source

To simplify the notation, we scale $\mathcal{C}_{10}$ to a unit power $\mathcal{C}_{10}^* = \Lambda_c^* \cap \mathcal{V}_m^*$, where $\sigma^2(\Lambda_m^*) = 1$.

During block b, the source sends

$$X_1(b) = \sqrt{\alpha P_1}[c_{10}^*(b-1) + u_m^*(b-1)] \bmod \Lambda_m^* + \sqrt{\bar{\alpha} P_1}[c_1(b) + u_s(b)] \bmod \Lambda_s,$$

where u_m^* and u_s are dithers uniformly distributed resp. over $\mathcal{V}_m^*$ and $\mathcal{V}_s$.

During block b, the relay sends

$$X_R(b) = \sqrt{P_R}[c_{10}^*(b-1) + u_m^*(b-1)] \bmod \Lambda_m^*,$$

where u_m^* is a dither uniformly distributed over $\mathcal{V}_m^*$.

- Decoding:
  During block b, the relay receives

$$Y_R(b) = h_{SR}\left(\sqrt{\alpha P_1}[c_{10}^*(b-1) + u_m^*(b-1)] \bmod \Lambda_m^* + \sqrt{\bar{\alpha} P_1}[c_1(b) + u_s] \bmod \Lambda_s(b)\right) + Z_R(b).$$

It first starts by removing $[c_{10}^*(b-1) + u_m^*(b-1)] \bmod \Lambda_m^*$ (the part of the message it has already decoded in the previous block) and decodes $c_1(b)$ which is possible if

$$R_1 \leq C\left(\frac{h_{SR}^2 \bar{\alpha} P_1}{N_R}\right).$$

At block b, the destination receives

$$Y_D(b) = \left(h_{SD}\sqrt{\alpha P_1} + h_{RD}\sqrt{P_R}\right)[c_{10}^*(b-1) + u_m^*(b-1)] \bmod \Lambda_m^*$$
$$+ h_{SD}\sqrt{\bar{\alpha} P_1}[c_1(b) + u_s(b)] \bmod \Lambda_s + Z_D(b).$$

It starts by decoding $c_{10}^*(b-1)$ which is possible if

$$R_{10} \leq C\left(\frac{\alpha h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\sqrt{\alpha P_1 P_R}}{h_{SD}^2 \bar{\alpha} P_1 + N_D}\right).$$

Then, it decodes $c_{11}(b-1)$ from the previous block which is possible if

$$R_{11} \leq C\left(\frac{h_{SD}^2 \bar{\alpha} P_1}{N_D}\right).$$

Thus, the decoding of $c_1(b-1)$ succeeds if

$$R_1 \leq C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\sqrt{\alpha P_1 P_R}}{N_D}\right).$$

□

4.1.4 Amplify-and-Forward

In Amplify-and-Forward (AF), the relay only scales its received message up to its power constraint, which is the most simple thing to do.

Proposition 4.1.4. *For the Gaussian relay channel, AF achieves the following rate:*

$$R_{AF} = \frac{1}{2}\log_2\left(\frac{\alpha + \sqrt{\alpha^2 - \beta^2}}{2}\right), \text{ where}$$

$$\alpha = 1 + \frac{h_{SD}^2 P_1 (h_{SR}^2 P_1 + N_R) + h_{SR}^2 h_{RD}^2 P_1 P_R}{h_{RD}^2 P_R N_R + h_{SR}^2 P_1 N_D + N_R N_D} \quad \text{and} \quad \beta = \frac{2h_{SD}h_{SR}h_{RD}P_1\sqrt{P_R(h_{SR}^2 P_1 + N_R)}}{h_{RD}^2 P_R N_R + h_{SR}^2 P_1 N_D + N_R N_D}. \tag{4.3}$$

Proof. The proof is based on [Chang et al., 2010] and on a block Markov encoding scheme.

During block b, the relay sends

$$X_R(b) = \sqrt{\frac{P_R}{h_{SR}^2 P_1 + N_R}}\big(h_{SR}X_1(b-1) + Z_R(b-1)\big).$$

At block b, the destination receives

$$Y_D(b) = \sqrt{\frac{P_R}{h_{SR}^2 P_1 + N_R}}h_{RD}h_{SR}X_1(b-1) + h_{SD}X_1(b) + \sqrt{\frac{P_R}{h_{SR}^2 P_1 + N_R}}h_{RD}Z_R(b-1) + Z_D.$$

The total noise power is: $N_{eq} = \dfrac{h_{RD}^2 P_R N_R}{h_{SR}^2 P_1 + N_R} + N_D$. We can divide $Y_D(b)$ by $\sqrt{N_{eq}}$ to obtain

$$\tilde{Y}_i(b) = \frac{\sqrt{h_{SR}^2 P_1 + N_R}h_{SD}}{\sqrt{h_{RD}^2 P_R N_R + (h_{SR}^2 P_1 + N_R)N_D}}X_1(b) + \frac{h_{RD}h_{SR}\sqrt{P_R}}{\sqrt{h_{RD}^2 P_R N_R + (h_{SR}^2 P_1 + N_R)N_D}}X_1(b-1) + Z_{eq}(b),$$

where $Z_{eq}(b)$ has unit power.

Thus, the AF protocol transforms the channel into a unit-memory intersymbol channel, as mentioned in [Kramer et al., Sept. 2005].
The achievable rate is then given by $R = \frac{1}{2}\frac{1}{2\pi}\int_0^{2\pi} log_2(1 + P_1|H(\omega)|^2)d\omega$, where $H(\omega)$ is the Fourier transform of $H = \left[\frac{\sqrt{h_{SR}^2 P_1 + N_R}h_{SD}}{\sqrt{h_{RD}^2 P_R N_R + (h_{SR}^2 P_1 + N_R)N_D}} \quad \frac{h_{RD}h_{SR}\sqrt{P_R}}{\sqrt{h_{RD}^2 P_R N_R + (h_{SR}^2 P_1 + N_R)N_D}}\right]$ and

$$|H(\omega)|^2 = \frac{h_{SD}^2(h_{SR}^2 P_1 + N_R) + h_{SR}^2 h_{RD}^2 P_R}{h_{RD}^2 P_R N_R + (h_{SR}^2 P_1 + N_R)N_D} + 2\frac{h_{SD}h_{SR}h_{RD}\sqrt{P_R(h_{SR}^2 P_1 + N_R)}}{h_{RD}^2 P_R N_R + (h_{SR}^2 P_1 + N_R)N_D}\cos(\omega).$$

Thus,

$$\frac{1}{2\pi}\int_0^{2\pi} log_2(1 + P_1|H(\omega)|^2)d\omega = \log_2\left(\frac{\alpha + \sqrt{\alpha^2 - \beta^2}}{2}\right)$$

with α and β given by (4.3) obtained from $\int_0^{2\pi} log_2(x + y\cos(z))dz = 2\pi \log_2\left(\frac{x+\sqrt{x^2-y^2}}{2}\right)$, [Gradshteyn and Ryzhik, 2007, 4.224.9]. □

4.1.5 Comparison of the presented protocols

In this subsection we compare the achievable rates for the full-duplex Gaussian relay channel using CF, DF and AF in a free space path loss model. We thus assume that the gains are inversely proportional to the distance to the power 3/2 between nodes. We suppose that the source and destination are unit distance apart and that the relay is between the source and destination at a distance d from the source: The channel gains are given as: $h_{SD} = 1$, $h_{SR} = 1/d^{3/2}$ and $h_{RD} = 1/(1-d)^{3/2}$.

For this numerical example, we set $P_1 = P_R = 10$ and $N_R = N_D = 1$.

Figure 4.7 gives the achievable rates using CF, AF and DF, as well as the cut-set bound as a function of the distance between the relay and the source.

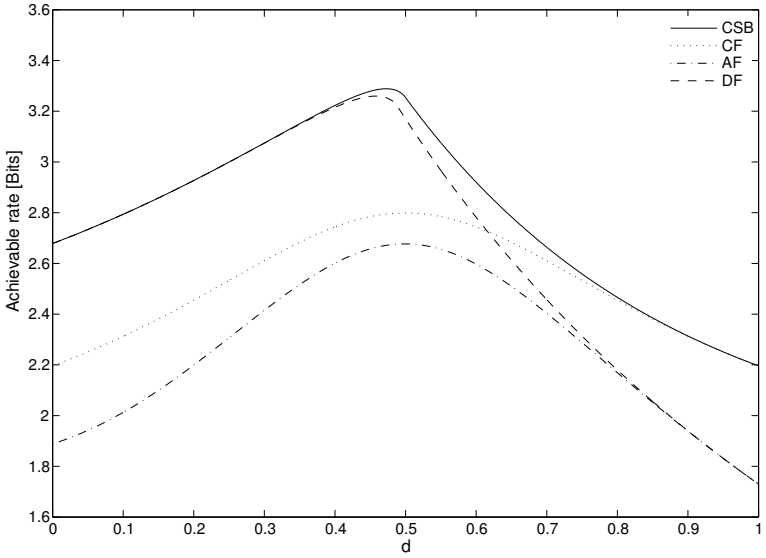


Figure 4.7: Comparison of the cut-set bound, DF, AF and CF as a function of the distance d between the source and the relay ($h_{SR} = 1/d^{3/2}$, $h_{RD} = 1/(1-d)^{3/2}$).

We can note that, when the relay is close to the source, DF outperforms both CF and AF, and is capacity achieving. When the relay is close to the destination, CF outperforms both DF

and AF, and is capacity achieving. When the relay is close to the source, the signal-to-noise ratio at the relay is very high, thus the relay can decode the transmitted signal at high rate, whereas when it's close to the destination, the decoding of the transmitted message is only possible at a very low rate. If the relay is close to the destination, the destination can easily recover the message sent by the relay, thus using CF, where the relay message is correlated with the source message, can achieve higher rates as DF. We can note that using this setup, AF always yields the worst performance in terms of achievable rate, but this protocol remains the simplest to implement at the relay.

4.2 Full-duplex Gaussian relay channel with correlated noises

In this section, we consider a more general Gaussian relay channel, where the additive Gaussian noises at the relay and the destination are correlated. This situation can occur for instance when a common interference signal contributes to the noises at both receivers. We will see that some relaying schemes can exploit this correlation. Since DF decodes the source message entirely, it removes all the correlation that could have been useful in terms of rate. On the other hand, CF looses some information via compression. [Zhang et al., March 2011] showed that none of the two strategies is optimal for all channel gains and correlation coefficient values. Nevertheless, they showed that for some specific values of the correlation coefficient, DF or CF is capacity-achieving.

The main contribution in this section is to prove that lattices can achieve the CF rate obtained theoretically in [Zhang et al., March 2011]. Part of this work was presented in [Savard and Weidmann, 2015b].

In the Gaussian case, the source sends X_1 of power P_1 and the relay X_R of power P_R. The received signals are given by:

$$\text{At the relay: } Y_R = h_{SR}X_1 + Z_R,$$
$$\text{At the destination: } Y_D = h_{SD}X_1 + h_{RD}X_R + Z_D,$$

where Z_D and Z_R are correlated Gaussian noises of variance N_D and N_R, respectively, and the correlation coefficient is defined as $\rho_z = \dfrac{\mathbb{E}[Z_D Z_R]}{\sqrt{N_D N_R}}$.

Since the capacity of this channel remains unknown, we first derive an upper bound, using a cut-set argument.

4.2.1 Cut-set bound

Proposition 4.2.1. *([Zhang et al., March 2011]) For the Gaussian relay channel with correlated noises, the CSB is given by*

$$R_{CSB} = \max_{\rho\in[0,1]} \min\left\{ C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\rho\sqrt{P_1 P_R}}{N_D}\right), \right.$$
$$\left. C\left(\frac{P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R - 2h_{SR}h_{SD}\rho_z\sqrt{N_R N_D})}{N_R N_D \overline{\rho_z^2}}\right)\right\}.$$

Proof. The proof follows the same arguments as for the Gaussian relay channel without correlated noises. The cut-set region is given by

$$R \le \min[I(X_1, X_R; Y_D), I(X_1; Y_D, Y_R|X_R)]. \tag{4.4}$$

We introduced the correlation coefficient: $\rho = \frac{\mathbb{E}[X_1 X_R]}{\sqrt{P_1 P_R}}$.

The computation of $I(X_1, X_R; Y_D)$ is done as in Section 4.1.1 and yields

$$I(X_1, X_R; Y_D) \leq C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\rho\sqrt{P_1 P_R}}{N_D}\right).$$

We write the second mutual information term of (4.4) as

$$\begin{aligned} I(X_1; Y_D, Y_R|X_R) &= H(Y_R, Y_D|X_R) - H(Y_R, Y_D|X_1, X_R) \\ &= H(Y_R, Y_D|X_R) - \frac{1}{2}\log_2\left((2\pi e)^2 N_R N_D(1-\rho_z^2)\right) \\ &= H(Y_D|X_R) + H(Y_R|Y_D, X_R) - \frac{1}{2}\log_2\left((2\pi e)^2 N_R N_D \overline{\rho_z^2}\right). \end{aligned} \tag{4.5}$$

Using the linear MMSE estimate of Y_D given X_R, we obtain

$$H(Y_D|X_R) \leq \frac{1}{2}\log_2\left(2\pi e(N_D + h_{SD}^2 P_1 \overline{\rho^2})\right).$$

Using the linear MMSE estimate of Y_R given Y_D and X_R we obtain

$$H(Y_R|Y_D, X_R) \leq \frac{1}{2}\log_2\left(2\pi e\left(\frac{N_R N_D \overline{\rho_z^2} + P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R - 2h_{SR}h_{SD}\rho_z\sqrt{N_R N_D})}{N_D + h_{SD}^2 P_1 \overline{\rho^2}}\right)\right).$$

Finally, inserting these bounds in (4.5), we obtain

$$I(X_1; Y_D, Y_R|X_R) \leq C\left(\frac{P_1\overline{\rho^2}(h_{SR}^2 N_D + h_{SD}^2 N_R - 2h_{SR}h_{SD}\rho_z\sqrt{N_R N_D})}{N_R N_D \overline{\rho_z^2}}\right).$$

□

The remainder of the section presents the CF protocol adapted for this channel model, as well as a comparison of rates obtained with CF and DF.

4.2.2 Compress-and-Forward

Proposition 4.2.2. *([Savard and Weidmann, 2015b]) For the Gaussian relay channel with correlated noises, CF based on lattice coding achieves the following rate:*

$$R_{CF} = C\left(\frac{P_1}{N_D}\frac{h_{SD}^2(N_R + D) + h_{SR}^2 N_D - 2h_{SD}h_{SR}\rho_z\sqrt{N_D N_R}}{N_R\overline{\rho_z^2} + D}\right), \textit{where}$$

$$D = \frac{(h_{SR}^2 N_D + h_{SD}^2 N_R)P_1 + N_D N_R \overline{\rho_z^2} - 2h_{SR}h_{SD}P_1\rho_z\sqrt{N_D N_R}}{h_{RD}^2 P_R}.$$

Remark: This rate region is the same as obtained theoretically in Proposition 5 of [Zhang et al., March 2011].

Proof. The encoding and decoding procedure is based on block Markov coding and follows the encoding/decoding scheme used for CF in Section 4.1.2.

- Encoding:

  The codebook for the sender is given by $\mathcal{C}_1 = \Lambda_{c1} \cap \mathcal{V}_1$, where $\Lambda_1 \subset \Lambda_{c1}$ and Λ_1 is both Rogers- and Poltyrev-good and Λ_{c1} is Poltyrev-good. To ensure the power constraints, we choose $\sigma^2(\Lambda_1) = P_1$ and Λ_{c1} such that $|\mathcal{C}_1| = R_{CF}$.

  During block b, the source sends $c_1(b) \in \mathcal{C}_1$ as

  $$X_1(b) = [c_1(b) + u_1(b)] \bmod \Lambda_1,$$

  where u_1 is a dither uniformly distributed over $\mathcal{V}_1$.

  The quantization codebook at the relay is given by $\mathcal{C}_q = \{\Lambda_{cQ} \cap \mathcal{V}_Q\}$, where $\Lambda_Q \subseteq \Lambda_{cQ}$ and Λ_{cQ} is Rogers-good and Λ_Q is Poltyrev-good. We choose the quantization distortion $\sigma^2(\Lambda_{cQ}) = D$ and

  $$\sigma^2(\Lambda_Q) = h_{SR}^2 P_1 + N_R + D - \frac{(h_{SD}h_{SR}P_1 + \rho_z\sqrt{N_D N_R})^2}{h_{SD}^2 P_1 + N_D}. \tag{4.6}$$

  The noise correlation appears in the second moment of Λ_Q since this lattice is used to obtained the noisy version of the signal received at the relay, using the signal sent over the direct link as side information. By setting ρ_z to zero, (4.6) reduces to the second moment of Λ_Q used for CF on the Gaussian relay channel (4.1). The quantization rate is thus $R_q = \frac{1}{2}\log_2\left(\frac{\sigma^2(\Lambda_Q)}{D}\right)$.

  The codebook for the relay is given by $\mathcal{C}_R = \{\Lambda_{cR} \cap \mathcal{V}_R\}$, where $\Lambda_R \subseteq \Lambda_{cR}$ and Λ_R is both Rogers- and Poltyrev-good and Λ_{cR} is Poltyrev-good. To ensure the power constraint, we choose $\sigma^2(\Lambda_R) = P_R$. Each compression index $i \in \mathcal{C}_q$ is mapped to one codeword $c_R \in \mathcal{C}_R$, that is Λ_R is chosen s.t. $|\mathcal{C}_R| = R_q$.

  During block b, the relay sends

  $$X_R(b) = [c_R(I(b-1)) + u_R(b)] \bmod \Lambda_R,$$

  where u_R is a dither uniformly distributed over $\mathcal{V}_R$.

- Decoding:

  During block b, the relay receives $Y_R(b) = h_{SR}X_1(b) + Z_R(b)$ and quantizes it to

  $$\begin{aligned} I(b) &= [Q_{cQ}(h_{SR}X_1(b) + Z_R(b) + u_{cQ}(b))] \bmod \Lambda_Q \\ &= [h_{SR}X_1(b) + Z_R(b) + u_{cQ}(b) - E_{cQ}(b)] \bmod \Lambda_Q, \end{aligned}$$

  where E_{cQ} is the quantization error and u_{cQ} is a dither uniformly distributed over $\mathcal{V}_{cQ}$.

  During block b, the destination receives

  $$Y_D(b) = h_{SD}X_1(b) + h_{RD}[c_R(I(b-1)) + u_R(b)] \bmod \Lambda_R + Z_D(b).$$

  It starts by decoding the quantization index, considering the source signal as noise, which is possible if

  $$R_q \leq C\left(\frac{h_{RD}^2 P_R}{h_{SD}^2 P_1 + N_D}\right).$$

Then, it forms $\widetilde{Y}_D(b) = h_{SD}X_1(b) + Z_D(b)$ by subtracting the relay signal.

The decoding of $X_1(b-1)$ is performed using Wyner-Ziv coding. During the previous block, the destination formed $\widetilde{Y}_D(b-1)$ which is used in block b as side information to estimate $\hat{Y}_R(b-1)$, a noisy version of the signal received at the relay:

$$\begin{aligned}\hat{Y}_R(b-1) &= \Big[I(b-1) - u_{cQ}(b-1) - \beta\widetilde{Y}_D(b-1)\Big] \bmod \Lambda_Q + \beta\widetilde{Y}_D(b-1)\\ &= \Big[h_{SR}X_1(b-1) + Z_R(b-1) - E_{cQ}(b-1) - \beta\widetilde{Y}_D(b-1)\Big] \bmod \Lambda_Q + \beta\widetilde{Y}_D(b-1)\\ &= h_{SR}X_1(b-1) + Z_R(b-1) - E_{cQ}(b-1).\end{aligned}$$

The last equality is valid under perfect decoding, requiring

$$\sigma^2(\Lambda) \geq (h_{SR} - \beta h_{SD})^2 P_1 + N_R + D + \beta^2 N_D - 2\beta\rho_z\sqrt{N_D N_R}.$$

Since we estimate $h_{SR}X_1(b-1) + Z_R(b-1)$ from $h_{SD}X_1(b-1) + Z_D(b-1)$, the linear MMSE orthogonality principle requires that β is chosen as $\beta = \dfrac{h_{SD}h_{SR}P_1 + \rho_z\sqrt{N_D N_R}}{h_{SD}^2 P_1 + N_D}$.

Thus,

$$\sigma^2(\Lambda) = h_{SR}^2 P_1 + N_R + D - \frac{(h_{SD}h_{SR}P_1 + \rho_z\sqrt{N_D N_R})^2}{h_{SD}^2 P_1 + N_D}.$$

Combining this with the quantization rate constraint, the distortion of the quantizer (in other words $\sigma^2(\Lambda_{cQ})$) is

$$D = \frac{(h_{SR}^2 N_D + h_{SD}^2 N_R)P_1 + N_D N_R\overline{\rho_z^2} - 2h_{SR}h_{SD}P_1\rho_z\sqrt{N_D N_R}}{h_{RD}^2 P_R}. \tag{4.7}$$

Note that by setting the correlation coefficient to zero, (4.7) reduces to the distortion obtained for the Gaussian relay channel without correlated noises, given by (4.2).

In order to recover $X_1(b-1)$, the receiver coherently uses two correlated noisy observations of $X_1(b-1)$ ($\hat{Y}_R(b-1)$ and $\widetilde{Y}_D(b-1)$) as

$$\begin{aligned}&\left(\frac{h_{SR}\sqrt{P_1}}{N_R + D} - \frac{h_{SD}\sqrt{P_1}\rho_z\sqrt{N_D N_R}}{N_D(N_R + D)}\right)\hat{Y}_R(b-1) + \left(\frac{h_{SD}\sqrt{P_1}}{N_D} - \frac{h_{SR}\sqrt{P_1}\rho_z\sqrt{N_D N_R}}{N_D(N_R + D)}\right)\widetilde{Y}_D(b-1)\\ &= X_1(b-1)\left(\frac{\left(h_{SD}^2(N_R + D) + h_{SR}^2 N_D - 2h_{SD}h_{SR}\rho_z\sqrt{N_D N_R}\right)\sqrt{P_1}}{N_D(N_R + D)}\right)\\ &+ Z_D(b-1)\left(\frac{\left(h_{SD}(N_R + D) - h_{SR}\rho_z\sqrt{N_D N_R}\right)\sqrt{P_1}}{N_D(N_R + D)}\right)\\ &+ (Z_R(b-1) - E_{cQ}(b-1))\left(\frac{\left(h_{SR}N_D - h_{SD}\rho_z\sqrt{N_D N_R}\right)\sqrt{P_1}}{N_D(N_R + D)}\right).\end{aligned}$$

Thus, decoding succeeds if

$$R_{CF} \leq C\left(\frac{P_1}{N_D}\frac{h_{SD}^2(N_R + D) + h_{SR}^2 N_D - 2h_{SD}h_{SR}\rho_z\sqrt{N_D N_R}}{N_R\overline{\rho_z^2} + D}\right).$$

□

This rate can also be achieved by an equivalent Gaussian relay channel with independent noises, by setting the source-relay link to $h'_{SR} = \left| h_{SR} - h_{SD}\rho_z\sqrt{\frac{N_R}{N_D}} \right|$ and the noise power at the relay to $N'_R = N_R\overline{\rho_z^2}$. When $\rho_z = \mp 1$, the relay is noiseless and CF achieves the rate $C\left(\frac{h_{SD}^2 P_1}{N_D} + \frac{h_{RD}^2 P_R}{N_D}\right)$. Moreover, using this equivalent model, CF degrades to direct transmission when $\rho_z = \sqrt{\frac{N_D}{N_R}}\frac{h_{SR}}{h_{SD}}$.

4.2.3 Decode-and-Forward

Since in DF the relay decodes the message, the fact that the noises are correlated doesn't affect the set of achievable rates.

Proposition 4.2.3. *([Nokleby and Aazhang, 2011]) For the Gaussian relay channel with correlated noises, DF achieves the same rate as an ordinary Gaussian relay channel (see Proposition 4.1.3):*

$$R_{DF} = \max_{\alpha\in[0,1]} \min\left\{ C\left(\frac{h_{SR}^2\bar{\alpha}P_1}{N_R}\right), C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\sqrt{\alpha P_1 P_R}}{N_D}\right)\right\}.$$

4.2.4 Capacity-achieving special cases

In [Zhang et al., March 2011], the authors proved that for the case when $N_R = N_D = N$, DF and CF can achieve capacity for some particular values of the correlation coefficient. In fact, these two specific values of the noise correlation coefficient correspond to the degraded Gaussian relay channel and the reversely-degraded Gaussian relay channel defined in [Cover and El Gamal, Sept. 1979].

Proposition 4.2.4. *([Zhang et al., March 2011]) For the Gaussian relay channel with correlated noises and $N_R = N_D = N$, DF achieves the capacity for $\rho_z = \frac{h_{SD}}{h_{SR}}$ and the capacity is given by:*

$$C = \max_{\alpha\in[0,1]} \min\left\{ C\left(\frac{h_{SR}^2\bar{\alpha}P_1}{N}\right), C\left(\frac{h_{SD}^2 P_1 + h_{RD}^2 P_R + 2h_{SD}h_{RD}\sqrt{\alpha P_1 P_R}}{N}\right)\right\}.$$

The capacity-achieving case, by setting this specific value to the correlation coefficient, is equivalent to the degraded Gaussian relay channel.

Proposition 4.2.5. *([Zhang et al., March 2011]) For the Gaussian relay channel with correlated noises and $N_R = N_D = N$, CF achieves the capacity for $\rho_z = \frac{h_{SR}}{h_{SD}}$ and the capacity is given by:*

$$C = \left(\frac{h_{SD}^2 P_1}{N}\right).$$

The capacity-achieving case, by setting this specific value to the correlation coefficient, is equivalent to the conversely-degraded Gaussian relay channel. Moreover, in this case, the capacity is achieved using the direct transmission, which is a special case of CF.

The proofs of these two propositions can be found in [Zhang et al., March 2011].

4.2.5 Comparison of the proposed protocols

We study the achievable rates using CF and DF for the full-duplex Gaussian relay channel with correlated noises, as a function of the noise correlation coefficient, for two channel configurations, as in [Zhang et al., March 2011].

For this numerical example, we set $P_1 = P_R = 1$ and $N_R = N_D = 1$.

First, assume that gains are inversely proportional to the distance between nodes. We assume that the distance between the source and destination equals 1 and that the relay lies between the source and the destination at a distance d from the source. Thus the channel gains are $h_{SD} = 1$, $h_{SR} = 1/d$, $h_{RD} = 1/(1-d)$. Figure 4.8 gives the obtained rates for $d = 0.2$, and Figure 4.9 for $d = 0.8$. We can see that when $d = 0.2$, DF outperforms CF for all values of ρ_z, since in this case the signal-to-noise ratio at the relay is very strong allowing DF to perform at high rate. For $d = 0.8$, CF outperforms DF for most values of ρ_z and we can note that both protocols perform very close to the upper bound. In this case, since the signal-to-noise ratio at the relay is low, it makes sense to exploit the noise correlation using CF. On both figures, we can see that DF achieves the cut-set upper bound for $\rho_z = \frac{h_{SD}}{h_{SR}} = d$.

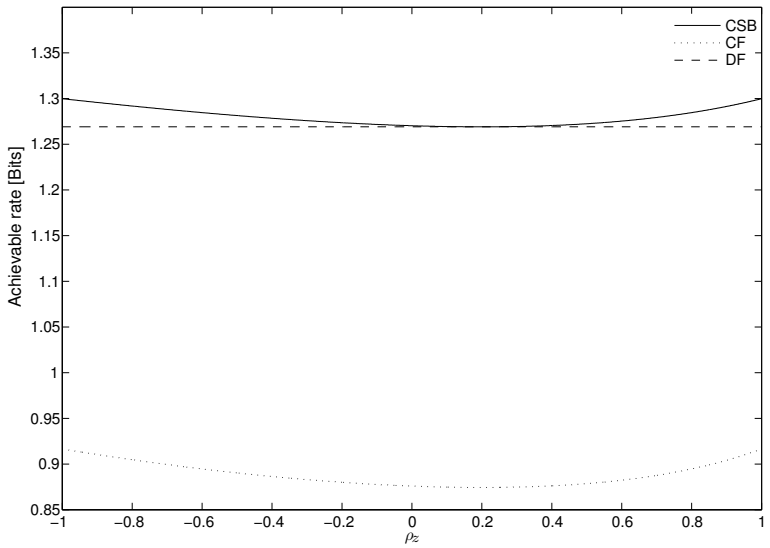


Figure 4.8: Comparison of the cut-set bound, DF and CF as a function of the noise correlation ρ_z ($h_{SD} = 1$, $h_{SR} = 1/d$, $h_{RD} = 1/(1-d)$ with $d = 0.2$).

For the second model, we assume that the relay is far from both the source and destination, such that the direct link has a higher gain than the relayed links. The channel gains are thus given as $h_{SD} = 1$, $h_{SR} = d$, $h_{RD} = 1 - d$. Figure 4.10 gives the obtained rates for $d = 0.2$, and Figure 4.11 for $d = 0.8$. In both cases, CF outperforms DF because the source-relay link is always weaker than the direct link, which strongly limits the performance of DF. Moreover, using only the direct link, a rate of 0.5 is achievable, which clearly outperform DF because of the weak source-relay link, compared to the source-destination link.

We can also see that CF achieves the upper-bound for $\rho_z = \frac{h_{SR}}{h_{SD}} = d$.

4.3 Conclusions

In this chapter, we presented a detailed study of the Gaussian relay channel, which is the most simple example of cooperative communication.

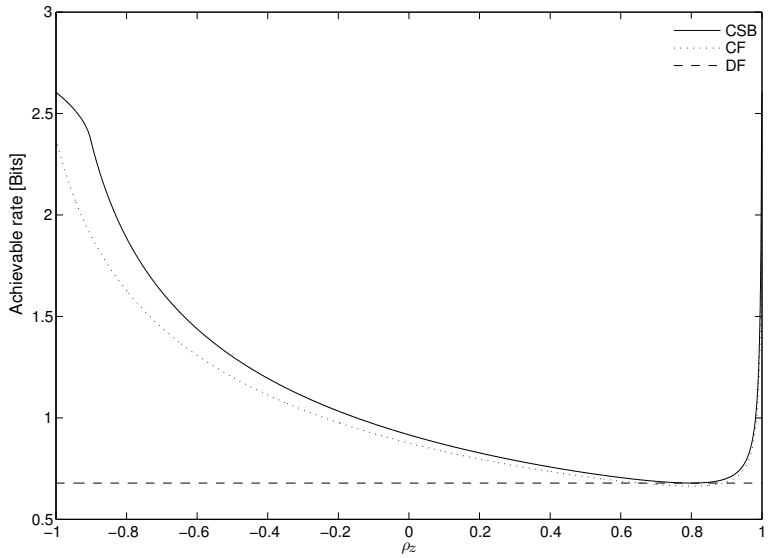


Figure 4.9: Comparison of the cut-set bound, DF and CF as a function of the noise correlation ρ_z ($h_{SD} = 1$, $h_{SR} = 1/d$, $h_{RD} = 1/(1-d)$ with $d = 0.8$).

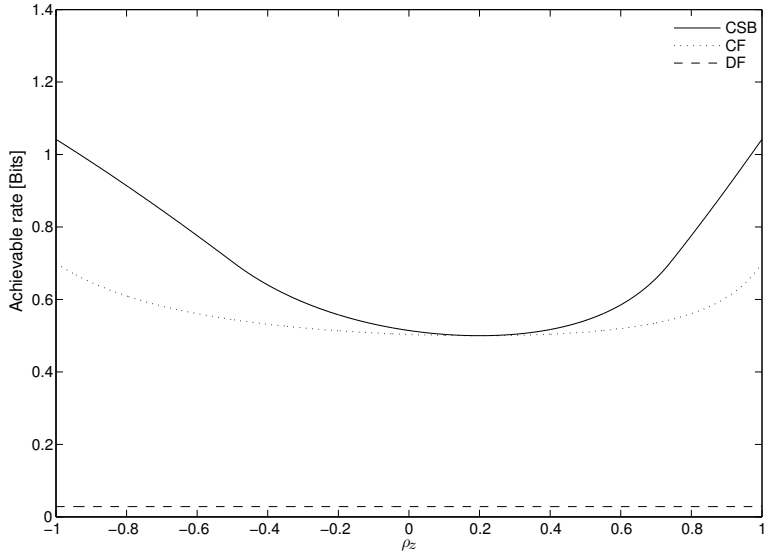


Figure 4.10: Comparison of the cut-set bound, DF and CF as a function of the noise correlation ρ_z ($h_{SD} = 1$, $h_{SR} = d$, $h_{RD} = 1 - d$ with $d = 0.2$).

We first gave an upper bound on the capacity as well as lower bounds achieved by various standard protocols such as DF, CF and AF using either lattice or AWGN coding. If the relay is between the source and the destination, we can see that if it is closer to the source, then DF outperforms CF and if the relay is closer to the destination, then CF outperforms DF.

In the second part of the chapter, we studied a more general Gaussian relay channel, where the additive Gaussian noises at the relay and destination are correlated. For this setup, we proved that lattices can achieve the CF rate. Again, neither DF nor CF give the best performance for all choices of channel gain and noise correlation. Nevertheless, the same observation as without noise correlation, based on the relay position, can be made. For some specific values of the correlation coefficient, CF or DF is capacity achieving.

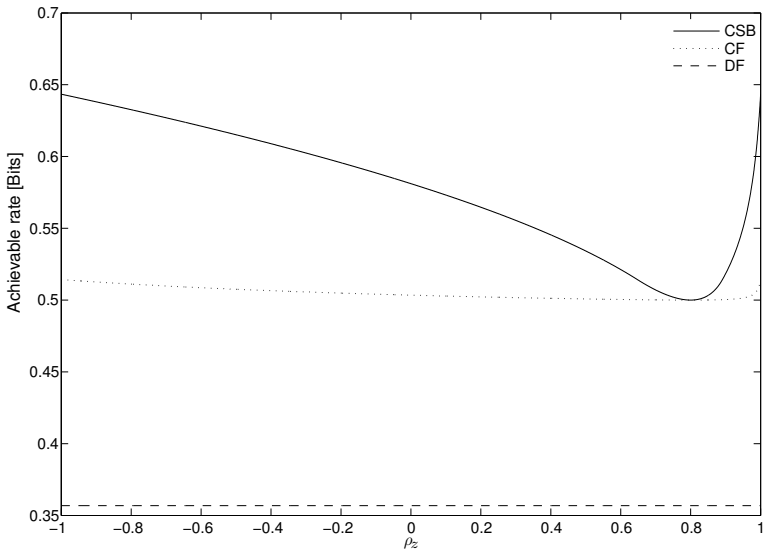


Figure 4.11: Comparison of the cut-set bound, DF and CF as a function of the noise correlation ρ_z ($h_{SD} = 1$, $h_{SR} = d$, $h_{RD} = 1 - d$ with $d = 0.8$).

Chapter 5

The Gaussian Two-way relay channel

Part of this work was presented in [Savard and Weidmann, 2015b].

Contents

The two-way relay channel (TWRC) is a natural extension of the relay channel, in which two users wish to exchange their messages with the help of one relay. Such a channel can be found in ad hoc networks or networks with a central node. A good understanding of this situation is a first step towards the understanding of multi-user information theory.

Throughout this chapter, we assume full duplex nodes (each node can transmit and receive at the same time) and restricted encoders: the channel input of each user depends only on its own message, and not on previously decoded ones.

The first two sections summarize achievable rates using standard schemes, such as DF, CF, AF or Compute-and-Forward (CoF), for two classes of two-way relay channels: with and without direct links. We focus again only on Gaussian channels: User 1 sends X_1 of power P_1, user 2 sends X_2 of power P_2 and the relay sends X_R of power P_R; noises at both users and the relay are Gaussian. The last section gives results for a more general TWRC with direct links and correlated noises at the relay and destinations.

5.1 Standard full-duplex Gaussian two-way relay channel without direct links

The two-way relay channel without direct links is depicted on Figure 5.1. In the Gaussian case, user 1 sends X_1 of power P_1, user 2 sends X_2 of power P_2 and the relay X_R of power P_R. The received signals are:

$$\begin{aligned} \text{At the relay: } Y_R &= g_{r1}X_1 + g_{r2}X_2 + Z_R; \\ \text{At user 1: } Y_1 &= g_{1r}X_R + Z_1; \\ \text{At user 2: } Y_2 &= g_{2r}X_R + Z_2, \end{aligned}$$

where Z_1, Z_2 and Z_R are Gaussian noises of variance N_1, N_2 and N_R respectively.

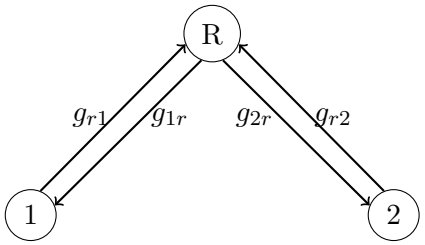


Figure 5.1: Two-way relay channel without direct links

5.1.1 Cut-set bound

As for the Gaussian relay channel, the capacity region of the Gaussian two-way relay channel remains unknown, thus we again start by upper bounding the capacity, using a cut-set argument.

Proposition 5.1.1. *([El Gamal and Kim, 2011]) For the Gaussian two-way relay channel without direct links, the CSB with restricted encoders is given by the convex closure of the cut-set region:*

$$\bigcup (R_1, R_2) : \left\{ R_1 \leq \min \left\{ C\left(\frac{g_{r1}^2 P_1}{N_R}\right), C\left(\frac{g_{2r}^2 P_R}{N_2}\right) \right\}, \right.$$
$$\left. R_2 \leq \min \left\{ C\left(\frac{g_{r2}^2 P_2}{N_R}\right), C\left(\frac{g_{1r}^2 P_R}{N_1}\right) \right\} \right\}.$$

Proof. This channel is the concatenation of point-to-point channels, thus the capacity is determined by the bottleneck link. □

In the following, we extensively use results on the Multiple Access Channel (MAC). Suppose now that many sources wish to transmit independent data to some specific destination node set. We focus here on a specific case where there is only one destination node that wishes to recover all messages sent. This network is called Multiple Access Channel (MAC). This situation occurs for example when K base stations communicate with a satellite. Questions as what rates are achievable simultaneously or how the sources should cooperate with each other arise in a very natural way.

Assume that the K transmitters have each a power constraint P and that the communication takes place over a Gaussian MAC. This situation is depicted on Figure 5.2. The received signal is $Y = \sum_{k=1}^{K} X_k + Z$, where Z is a Gaussian additive noise of variance N.

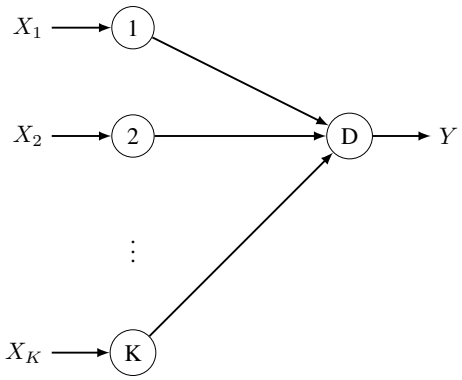


Figure 5.2: Example of a MAC with K users

Proposition 5.1.2. *([Cover and Thomas, 2006]) The following rate region is achievable for the Gaussian MAC with K users of power P:*

$$\begin{aligned} R_i &< \frac{1}{2}\log_2\left(1+\frac{P}{N}\right) \quad \forall i \\ R_i + R_j &< \frac{1}{2}\log_2\left(1+\frac{2P}{N}\right) \quad \forall i \neq j \\ \sum_{k=1}^{K} R_k &< \frac{1}{2}\log_2\left(1+\frac{KP}{N}\right). \end{aligned}$$

Note that if all rates are equal, the last inequality dominates the others.

5.1.2 Decode-and-Forward

Proposition 5.1.3. *([El Gamal and Kim, 2011]) For the Gaussian two-way relay channel without direct links, DF with restricted encoders achieves the following rate region:*

$$\begin{aligned} \bigcup (R_1, R_2) : \Bigg\{ R_1 &\leq \min\left\{ C\left(\frac{g_{r1}^2 P_1}{N_R}\right), C\left(\frac{g_{2r}^2 P_R}{N_2}\right)\right\}, \\ R_2 &\leq \min\left\{ C\left(\frac{g_{r2}^2 P_2}{N_R}\right), C\left(\frac{g_{1r}^2 P_R}{N_1}\right)\right\}, \\ R_1 + R_2 &\leq C\left(\frac{g_{r1}^2 P_1 + g_{r2}^2 P_2}{N_R}\right)\Bigg\}. \end{aligned}$$

Proof. The relay recovers the messages from both users over a MAC and broadcasts them back to the users. □

5.1.3 Amplify-and-Forward

Proposition 5.1.4. *([El Gamal and Kim, 2011]) For the Gaussian two-way relay channel without direct links, AF with restricted encoders achieves the following rate region:*

$$\begin{aligned} \bigcup (R_1, R_2) : \Bigg\{ R_1 &\leq C\left(\frac{g_{r2}^2 P_2 + g_{r1}^2 P_1 + N_R + g_{r1}^2 g_{2r}^2 P_1 P_R}{g_{2r}^2 P_R N_R + (g_{r2}^2 P_2 + g_{r1}^2 P_1 + N_R) N_2}\right), \\ R_2 &\leq C\left(\frac{g_{r1}^2 P_1 + g_{r2}^2 P_2 + N_R + g_{r2}^2 g_{1r}^2 P_2 P_R}{g_{1r}^2 P_R N_R + (g_{r1}^2 P_1 + g_{r2}^2 P_2 + N_R) N_1}\right)\Bigg\}. \end{aligned}$$

Proof. The relay sends a scaled version of its received message

$$X_R = \sqrt{\frac{P_R}{g_{r1}^2 P_1 + g_{r2}^2 P_2 + N_R}}(g_{r1}X_1 + g_{r2}X_2 + Z_R).$$

□

5.1.4 Compress-and-Forward

In [Smirani et al., Jun. 2014], the achievable rate region using CF was characterized for the half-duplex Gaussian relay channel. Here we present a full-duplex version of that protocol.

Proposition 5.1.5. *For the Gaussian two-way relay channel without direct links, when assuming w.l.o.g* $g_{r2}^2 P_2 \leq g_{r1}^2 P_1$*, CF with restricted encoders achieves the following rate region using lattice coding:*

$$\bigcup (R_1, R_2) : \left\{ \begin{aligned} R_1 \leq & C\left(\frac{g_{r1}^2 P_1 (g_{r1}^2 P_1 + N_R - D)}{(g_{r1}^2 P_1 + N_R - D)N_R + D(g_{r1}^2 P_1 + N_R)}\right), \\ R_2 \leq & C\left(\frac{g_{r2}^2 P_2 (g_{r1}^2 P_1 + N_R - D)}{(g_{r1}^2 P_1 + N_R - D)N_R + D(g_{r1}^2 P_1 + N_R)}\right) \end{aligned} \right\} \text{ with}$$

$$D = \frac{g_{r1}^2 P_1 + N_R}{1 + P_R \min\left\{\frac{g_{1r}^2}{N_1}, \frac{g_{2r}^2}{N_2}\right\}}.$$

Proof. The proof follows the lines of the half-duplex protocol in [Smirani et al., Jun. 2014]. Let $S_i = g_{ri}X_i$ be the side information available at user i and $U_i = g_{rj}X_j + Z_R$, $i \neq j \in \{1, 2\}$ denote the unknown part at user i. Since we assumed that $g_{r2}^2 P_2 \leq g_{r1}^2 P_1$, user 1 has the weakest side information, thus the compression step at the relay is performed such that user 1 reconstructs an estimate $\hat{Y}_{R,1}$ of Y_R with MSE distortion D.

- Encoding:

  For the quantization at the relay, we use a nested lattice coding scheme with $\Lambda_2 \subseteq \Lambda_1$, where Λ_2 is good for quantization and Λ_1 is good for channel coding. We set $\sigma^2(\Lambda_1) = D$ and $\sigma^2(\Lambda_2) = g_{r1}^2 P_1 + N_R$. Thus, the quantization rate is $R_q = \frac{1}{2}\log_2\left(\frac{g_{r1}^2 P_1 + N_R}{D}\right)$.

  The relay performs a Wyner-Ziv coding for which it outputs the index of

  $$v_R = [Q_1(\beta Y_R + t_1)] \bmod \Lambda_2,$$

  where t_1 is a dither uniformly distributed over $\mathcal{V}_1$.

- Decoding:

  At both users, v_R is decoded first and $\hat{U}_i$ is recovered using Wyner-Ziv decoding with S_i as side-information:

  $$\hat{U}_i = \beta\left([v_R - t_1 - \beta S_i] \bmod \Lambda_2\right) \quad i \in \{1, 2\}.$$

  v_R can be decoded at both users as long as

  $$R_q \leq \min\left\{C\left(\frac{g_{1r}^2 P_R}{N_1}\right), C\left(\frac{g_{2r}^2 P_R}{N_2}\right)\right\}.$$

At user 2, the unknown part is recovered as

$$\begin{aligned}\hat{U}_2 &= \beta[[Q_1(\beta U_2 + \beta S_2 + t_1)] \bmod \Lambda_2 - t_1 - \beta S_2] \bmod \Lambda_2 \\ &= \beta[\beta U_2 + E_q] \bmod \Lambda_2 \\ &\overset{(a)}{=} \beta(\beta U_2 + E_q) \\ &= \beta^2 g_{r1} X_1 + \beta^2 Z_R + \beta E_q\end{aligned}$$

where (a) is valid under perfect decoding, requiring

$$\sigma^2(\Lambda_2) \geq \beta^2(g_{r1}^2 P_1 + N_R) + D. \tag{5.1}$$

Decoding of X_1 succeeds if $R_1 \leq C\left(\frac{\beta^2 g_{r1}^2 P_1}{\beta^2 N_R + D}\right)$.

Let $\hat{Y}_{R,2}$ denote an estimate of Y_R obtained by user 2. Since

$$\begin{aligned}Y_R - \hat{Y}_{R,2} &= U_2 - \hat{U}_2 \\ &= (1-\beta^2)U_2 - \beta E_q,\end{aligned}$$

the MSE distortion equals $\mathbb{E}[(Y_R - \hat{Y}_{R,2})^2] = (1-\beta^2)^2(g_{r1}^2 P_1 + N_R) + \beta^2 D$, and must, as at user 1, satisfy the fidelity criterion,

$$\mathbb{E}[(Y_R - \hat{Y}_{R,2})^2] \leq D. \tag{5.2}$$

The optimal β satisfying both (5.1) and (5.2) is $\beta = \sqrt{1 - \frac{D}{g_{r1}^2 P_1 + N_R}}$.

The same decoding procedure is used at user 1: The unknown part is recovered as

$$\begin{aligned}\hat{U}_1 &= \beta[[Q_1(\beta U_1 + \beta S_1 + u_1)] \bmod \Lambda_2 - u_1 - \beta S_1] \bmod \Lambda_2 \\ &= \beta[\beta U_1 + E_q] \bmod \Lambda_2 \\ &\overset{(b)}{=} \beta(\beta U_1 + E_q) \\ &= \beta^2 g_{r2} X_2 + \beta^2 Z_R + \beta E_q\end{aligned}$$

where (b) is valid under perfect decoding, requiring

$$\sigma^2(\Lambda_2) \geq \beta^2(g_{r2}^2 P_2 + N_R) + D. \tag{5.3}$$

Thus, decoding of X_2 succeeds if $R_2 \leq C\left(\frac{\beta^2 g_{r2}^2 P_2}{\beta^2 N_R + D}\right)$.

Note that since $\beta = \sqrt{1 - \frac{D}{g_{r1}^2 P_1 + N_R}}$, the condition (5.3) is satisfied:

$$\begin{aligned}\beta^2(g_{r2}^2 P_2 + N_R) + D &= \beta^2(g_{r2}^2 P_2 + N_R) + g_{r1}^2 P_1 + N_R - \beta^2(g_{r1}^2 P_1 + N_R) \\ &= g_{r1}^2 P_1 + N_R + \beta^2(g_{r2}^2 P_2 - g_{r1}^2 P_1) \\ &\leq g_{r1}^2 P_1 + N_R \\ &= \sigma^2(\Lambda_2),\end{aligned}$$

where the last equality holds by design.

The fidelity criterion at user 1 is also satisfied since

$$\begin{aligned}\mathbb{E}[(Y_R - \hat{Y}_{R,1})^2] &= \mathbb{E}[(U_1 - \hat{U}_1)^2] \\ &= \mathbb{E}[((1-\beta^2)U_1 - \beta E_q)^2] \\ &= (1-\beta^2)^2(g_{r2}^2 2 + N_R) + \beta^2 D \\ &= D - \frac{D^2}{(g_{r1}^2 P_1 + N_R)^2}(g_{r1}^2 P_1 - g_{r2}^2 P_2) \\ &\overset{(c)}{\leq} D,\end{aligned}$$

where (c) follows from the assumption $g_{r1}^2 P_1 \geq g_{r2}^2 P_2$.

□

5.1.5 Comparison of the presented protocols

In this subsection we compare the achievable rates for the full-duplex Gaussian two-way relay channel using CF, DF and AF in a free space path loss model. We thus assume that the gains are inversely proportional to the distance between nodes raised to the power 3/2. We suppose that source and destination are unit distance apart and that the relay is between the source and the destination at a distance d from the source: The channel gains are given as: $g_{12} = g_{21} = 1$, $g_{1r} = g_{r1} = 1/d^{3/2}$ and $g_{r2} = g_{2r} = 1/(1-d)^{3/2}$. For this numerical example, we set $P = P_R = 10$ and $N_R = N_D = 1$.

Figure 5.3 gives the achievable sum rate of CF, DF and AF as well as the cut-set bound as a function of the distance d between the relay and user 1. We can see that DF performs very close to the upper bound when the relay is close to user 1 (or user 2) and that CF gives the best performance when the relay is somewhere in the middle. We can also note that when the relay is close to one user, AF achieves higher sum rate than CF, but lower than DF and when the relay is somewhere in the middle, AF achieves higher sum rate than DF, but lower than CF.

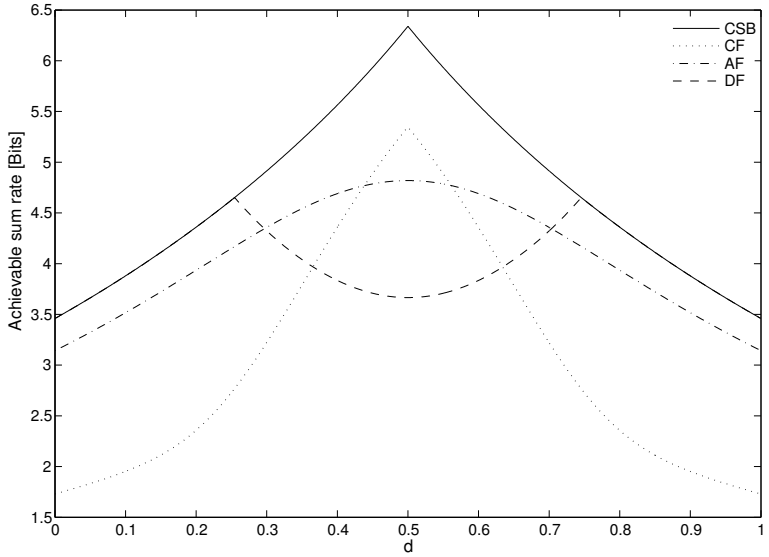


Figure 5.3: Comparison of the cut-set bound, DF, AF and CF as the function of the distance d between user 1 and the relay ($g_{1r} = g_{r1} = 1/d^{3/2}$, $g_{2r} = g_{r2} = 1/(1-d)^{3/2}$).

5.2 Standard full-duplex Gaussian two-way relay channel with direct links

The two-way relay channel with direct links is depicted on Figure 5.4. As previously, in the Gaussian case, user 1 sends X_1 of power P_1, user 2 X_2 of power P_2 and the relay X_R of power P_R. The received signals are:

$$\begin{aligned} \text{At the relay: } Y_R &= g_{r1}X_1 + g_{r2}X_2 + Z_R; \\ \text{At user 1: } Y_1 &= g_{12}X_2 + g_{1r}X_R + Z_1; \\ \text{At user 2: } Y_2 &= g_{21}X_1 + g_{2r}X_R + Z_2, \end{aligned}$$

where Z_1, Z_2 and Z_R are Gaussian noises of variance N_1, N_2 and N_R respectively.

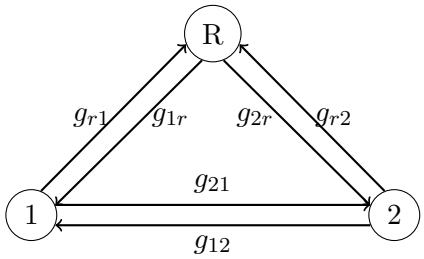


Figure 5.4: Two-way relay channel with direct links

5.2.1 Cut-set bound

As previously, the capacity region of this channel remains unknown, thus we first derive an upper bound based on a cut-set argument.

Proposition 5.2.1. *For the Gaussian two-way relay channel with direct links, the CSB with restricted encoders is given by the convex closure of the cut-set region:*

$$\bigcup_{0\leq\rho_1,\rho_2\leq 1} (R_1, R_2) : \Bigg\{$$

$$R_1 \leq \min\left\{ C\left(\frac{P_1(\overline{\rho_2^2}-\rho_1^2)}{\overline{\rho_2^2}}\frac{g_{r1}^2N_2 + g_{21}^2N_R}{N_2N_R}\right), C\left(\frac{g_{2r}^2P_R\overline{\rho_2^2} + g_{21}^2P_1 + 2g_{21}g_{2r}\rho_1\sqrt{P_1P_R}}{N_2}\right)\right\},$$

$$R_2 \leq \min\left\{ C\left(\frac{P_2(\overline{\rho_1^2}-\rho_2^2)}{\overline{\rho_1^2}}\frac{g_{r2}^2N_1 + g_{12}^2N_R}{N_1N_R}\right), C\left(\frac{g_{1r}^2P_R\overline{\rho_1^2} + g_{12}^2P_2 + 2g_{12}g_{1r}\rho_2\sqrt{P_2P_R}}{N_1}\right)\right\}\Bigg\}.$$

Proof. The cut-set region is given by:

$$\begin{cases} R_1 \leq \min[I(X_1; Y_R, Y_2|X_R, X_2), I(X_1, X_R; Y_2|X_2)], \\ R_2 \leq \min[I(X_2; Y_R, Y_1|X_R, X_1), I(X_2, X_R; Y_1|X_1)]. \end{cases} \tag{5.4}$$

In this proof, we only compute

$$R_1 \leq \min[I(X_1; Y_R, Y_2|X_R, X_2), I(X_1, X_R; Y_2|X_2)], \tag{5.5}$$

the bound on R_2 follows in analogous fashion.

We introduce two correlation coefficients: $\rho_1 = \frac{\mathbb{E}[X_1 X_R]}{\sqrt{P_1 P_R}}$ and $\rho_2 = \frac{\mathbb{E}[X_2 X_R]}{\sqrt{P_2 P_R}}$.

The first mutual information of (5.5) is written as

$$\begin{aligned} I(X_1; Y_R, Y_2 | X_R, X_2) &= H(Y_R, Y_2 | X_R, X_2) - H(Y_R, Y_2 | X_R, X_2, X_1) \\ &= H(Y_2 | X_2, X_R) + H(Y_R | Y_2, X_2, X_R) - H(Y_R, Y_2 | X_R, X_2, X_1). \end{aligned}$$

Using the linear MMSE estimate of Y_2 given X_2 and X_R, we obtain

$$H(Y_2 | X_2, X_R) \leq \frac{1}{2} \log_2 \left(2\pi e \left(\frac{g_{21}^2 P_1 \left(\rho_1^2 - \overline{\rho_2^2} \right)}{-\overline{\rho_2^2}} + N_2 \right) \right).$$

Using the linear MMSE estimate of Y_R given Y_2, X_2 and X_R, we obtain

$$H(Y_R | Y_2, X_2, X_R) \leq \frac{1}{2} \log_2 \left(2\pi e \left(\frac{g_{r1}^2 P_1 N_2 \left(\overline{\rho_2^2} - \rho_1^2 \right)}{g_{21}^2 P_1 \left(\overline{\rho_2^2} - \rho_1^2 \right) + N_2 \overline{\rho_2^2}} + N_R \right) \right).$$

Finally,

$$\begin{aligned} I(X_1; Y_R, Y_2 | X_R, X_2) &= H(Y_R, Y_2 | X_R, X_2) - H(Y_R, Y_2 | X_R, X_2, X_1) \\ &= H(Y_2 | X_2, X_R) + H(Y_R | Y_2, X_2, X_R) - H(Y_R, Y_2 | X_R, X_2, X_1) \\ &\leq C \left(\frac{P_1 (\overline{\rho_2^2} - \rho_1^2)}{\overline{\rho_2^2}} \frac{g_{r1}^2 N_2 + g_{21}^2 N_R}{N_2 N_R} \right). \end{aligned}$$

The second mutual information of (5.5) is written as

$$\begin{aligned} I(X_1, X_R; Y_2 | X_2) &= H(Y_2 | X_2) - H(Y_2 | X_1, X_2, X_R) \\ &\leq C \left(\frac{g_{2r}^2 P_R \overline{\rho_2^2} + g_{21}^2 P_1 + 2 g_{21} g_{2r} \rho_1 \sqrt{P_1 P_R}}{N_2} \right). \end{aligned}$$

□

The remaining of the section presents various protocols, such as DF, AF, Compute-and-Forward (CoF) and Compress/Decode-and-Forward (CDF).

5.2.2 Decode-and-Forward

Proposition 5.2.2. *([Rankov and Wittneben, 2006]) For the Gaussian two-way relay channel with direct links, DF with restricted encoders achieves the following rate region:*

$$\bigcup_{0 \leq \rho_1, \rho_2, \gamma \leq 1} (R_1, R_2) : \left\{ \begin{aligned} R_1 &\leq \min \left\{ C \left(\frac{g_{r1}^2 \overline{\rho_1^2} P_1}{N_R} \right), C \left(\frac{g_{21}^2 P_1 + \gamma g_{2r}^2 P_R + 2\rho_1 g_{21} g_{2r} \sqrt{\gamma P_1 P_R}}{N_2} \right) \right\}, \\ R_2 &\leq \min \left\{ C \left(\frac{g_{r2}^2 \overline{\rho_2^2} P_2}{N_R} \right), C \left(\frac{g_{12}^2 P_2 + \overline{\gamma} g_{1r}^2 P_R + 2\rho_2 g_{12} g_{1r} \sqrt{\overline{\gamma} P_2 P_R}}{N_1} \right) \right\}, \\ R_1 + R_2 &\leq C \left(\frac{g_{r1}^2 \overline{\rho_1^2} P_1 + g_{r2}^2 \overline{\rho_2^2} P_2}{N_R} \right) \end{aligned} \right\}.$$

ρ_1 and ρ_2 allow to trade off power between repeating the message from the previous block and sending a new one at user 1 and 2, respectively. γ controls the power trade-off between the messages intended for user 1 and 2.

Proof. The proof is based on sliding window encoding and decoding to take advantage of both the direct and the relayed links. This extends the proof proposed in [Khina et al., 2012] for the Gaussian relay channel to the Gaussian TWRC.

- Encoding:

  The codeword of each user is the superposition of two codewords. At the relay, only one of the two codewords is decoded and sent coherently to the receiver with the user message in the next block, the other codeword is decoded only at the receiver.

  At block b, user 1 sends

  $$X_1(b) = \sqrt{\frac{\rho_1^2 P_1}{P_R \gamma}} X_{11}(w_{1,b}) + \sqrt{\overline{\rho_1^2}} X_{12}(w_{1,b+1}),$$

  where X_{11} is of power $P_R\gamma$ and X_{12} of power P_1.

  At block b, user 2 sends

  $$X_2(b) = \sqrt{\frac{\rho_2^2 P_2}{P_R \overline{\gamma}}} X_{21}(w_{2,b}) + \sqrt{\overline{\rho_2^2}} X_{22}(w_{2,b+1}),$$

  where X_{21} is of power $P_R\overline{\gamma}$ and X_{22} of power P_2.

  The codebooks have rate R_1 and R_2 respectively:

  $$w_{1,b} \in \{1, \dots, 2^{nR_1}\} \text{ and } w_{2,b} \in \{1, \dots, 2^{nR_2}\}.$$

  At block b, the relay sends $X_R(b) = X_{11}(w_{1,b}) + X_{21}(w_{2,b})$.

- Decoding:

  At block b, the relay receives

  $$\begin{aligned} Y_R(b) =& g_{r1} \sqrt{\frac{\rho_1^2 P_1}{\gamma P_R}} X_{11}(w_{1,b}) + g_{r1} X_{12}(w_{1,b+1}) \\ &+ g_{r2} \sqrt{\frac{\rho_2^2 P_2}{\overline{\gamma} P_R}} X_{21}(w_{2,b}) + g_{r2} X_{22}(w_{2,b+1}) + Z_R(b). \end{aligned}$$

  During the previous block, the relay has decoded $w_{1,b}$ and $w_{2,b}$ (by induction assumption), so it can remove them and decode $w_{1,b+1}$ and $w_{2,b+1}$ if

  $$\begin{aligned} R_1 &\leq C\left(\frac{g_{r1}^2 \overline{\rho_1^2} P_1}{N_R}\right), \\ R_2 &\leq C\left(\frac{g_{r2}^2 \overline{\rho_2^2} P_2}{N_R}\right) \text{ and} \\ R_1 + R_2 &\leq C\left(\frac{g_{r1}^2 \overline{\rho_1^2} P_1 + g_{r2}^2 \overline{\rho_2^2} P_2}{N_R}\right). \end{aligned}$$

User 1 receives at block b:

$$Y_1(b) = \left(g_{12}\sqrt{\frac{\rho_2^2 P_2}{\overline{\gamma} P_R}} + g_{1r} \right) X_{21}(w_{2,b}) + g_{12}\sqrt{\overline{\rho_2^2}} X_{22}(w_{2,b+1}) + Z_1(b).$$

It starts decoding $X_{22}(w_{2,b})$ from $Y_1(b-1)$ and then decodes $X_{21}(w_{2,b})$ from $Y_1(b)$, considering $X_{22}(w_{2,b+1})$ as noise. This succeeds if

$$\begin{aligned} R_2 &\leq C\left(\frac{\left(g_{12}\sqrt{\frac{\rho_2^2 P_2}{\overline{\gamma} P_R}} + g_{1r} \right)^2 \overline{\gamma} P_R}{g_{12}^2 \overline{\rho_2^2} P_2 + N_1} \right) + C\left(\frac{g_{12}^2 \overline{\rho_2^2} P_2}{N_1} \right) \\ &= C\left(\frac{g_{12}^2 P_2 + \overline{\gamma} g_{1r}^2 P_R + 2\rho_2 g_{12} g_{1r} \sqrt{\overline{\gamma} P_2 P_R}}{N_1} \right). \end{aligned}$$

The same decoding procedure can be applied at user 2 and the decoding succeeds if

$$\begin{aligned} R_1 &\leq C\left(\frac{\left(g_{21}\sqrt{\frac{\rho_1^2 P_1}{\gamma P_R}} + g_{2r} \right)^2 \gamma P_R}{g_{12}^2 \overline{\rho_1^2} P_1 + N_2} \right) + C\left(\frac{g_{21}^2 \overline{\rho_1^2} P_1}{N_2} \right) \\ &= C\left(\frac{g_{21}^2 P_1 + \gamma g_{2r}^2 P_R + 2\rho_1 g_{21} g_{2r} \sqrt{\gamma P_1 P_R}}{N_2} \right). \end{aligned}$$

□

5.2.3 Amplify-and-Forward

Proposition 5.2.3. *For the Gaussian two-way relay channel with direct links, AF with restricted encoders achieves the following rate region:*

$$\bigcup (R_1, R_2) : \left\{ R_1 \leq \frac{1}{2}\log_2\left(\frac{\alpha_1 + \sqrt{\alpha_1^2 - \beta_1^2}}{2} \right), R_2 \leq \frac{1}{2}\log_2\left(\frac{\alpha_2 + \sqrt{\alpha_2^2 - \beta_2^2}}{2} \right) \right\}, \text{ where}$$

$$\begin{aligned} g &= \sqrt{\frac{P_R}{g_{r1}^2 P_1 + g_{r2}^2 P_2 + N_R}}; \\ \alpha_1 &= 1 + P_1 \frac{g_{21}^2 + g_{r1}^2 g_{2r}^2 g^2}{N_2 + g^2 N_R} \quad \text{and} \quad \beta_1 = \frac{2 g_{r1} g_{2r} g_{21} g P_1}{N_2 + g^2 N_R}; \\ \alpha_2 &= 1 + P_2 \frac{g_{12}^2 + g_{r2}^2 g_{1r}^2 g^2}{N_1 + g^2 N_R} \quad \text{and} \quad \beta_2 = \frac{2 g_{r2} g_{1r} g_{12} g P_2}{N_1 + g^2 N_R}. \end{aligned}$$

Proof. The proof follows the same approach as for AF in Section 4.1.4. □

5.2.4 Compute-and-Forward

In Compute-and-Forward (CoF), the relay decodes a linear combination of the transmitted codewords, by exploiting the noisy linear combination output by the channel. Using nested lattices, it has been shown that decoding linear combinations can be done at higher rates than decoding each codeword individually [Nazer and Gastpar, Oct. 2011].

For this scheme, we assume that $g_{r1} = g_{1r} = g_{r2} = g_{2r} = 1$. If the coefficients aren't equal to 1, one must find the best integer equation to compute at the relay.

Proposition 5.2.4. *([Song and Devroye, Aug. 2013]) For the Gaussian two-way relay channel with direct links, CoF with restricted encoders achieves the following rate region using lattice coding:*

$$\bigcup (R_1, R_2) : \left\{ R_1 \leq \min \left\{ \frac{1}{2} \log_2^+ \left(\frac{P_1}{P_1 + P_2} + \frac{P_1}{N_R} \right), C\left(\frac{g_{21}^2 P_1 + P_R}{N_2} \right) \right\}, \right.$$
$$\left. R_2 \leq \min \left\{ \frac{1}{2} \log_2^+ \left(\frac{P_2}{P_1 + P_2} + \frac{P_2}{N_R} \right), C\left(\frac{g_{12}^2 P_2 + P_R}{N_1} \right) \right\} \right\}.$$

5.2.5 Compress/Decode-and-Forward

It can be shown that for the Gaussian relay channel, DF approaches the cut-set bound when the relay is close to the user, while CF approaches the cut-set bound when the relay is close to the destination. For the Gaussian two-way relay channel, neither DF nor CF achieve the cut-set bound, since one of the rates obtained with these two protocols is always very low. Instead of using either DF or CF, one can combine them to achieve better performance. Assume, without loss of generality, that the relay is close to user 1: the relay first decodes the message from user 1, which can be done with a high rate, and then compresses the rest, which is also done with a high rate, since this compressed version is intended for user 1, that is close to the relay.

In this subsection, we assume without loss of generality that $g_{r2} P_2 \leq g_{r1} P_1$.

Proposition 5.2.5. *For the Gaussian two-way relay channel with direct links, Compress/Decode-and-Forward (CDF) with restricted encoders achieves the following rate region:*

$$\bigcup_{\substack{0 \leq \alpha \leq 1 \\ 0 < \gamma < 1}} (R_1, R_2) : \left\{ R_1 \leq \min \left\{ C\left(\frac{g_{r1}^2 \bar{\alpha} P_1}{g_{r2}^2 P_2 + N_R} \right), \frac{1}{2} \log_2 \left(\frac{g_{21}^2 P_1 + g_{2r}^2 P_R + N_2 + 2 g_{21} g_{2r} \sqrt{\alpha \bar{\gamma} P_1 P_R}}{g_{2r}^2 \gamma P_R + N_2} \right) \right\}, \right.$$
$$\left. R_2 \leq C\left(\frac{P_2}{N_1} \frac{g_{12}^2 (N_R + D) + g_{r2}^2 N_1}{N_R + D} \right) \right\} \text{ with } D = \frac{(g_{r2}^2 N_1 + g_{12}^2 N_R) P_2 + N_R N_1}{g_{1r}^2 \gamma P_R}.$$

α allows to trade off power between repeating the message from the previous block and sending a new one at user 1. γ controls the power trade-off between the messages intended for user 1 and 2.

Proof. For user 1, we use the DF scheme presented in Section 4.1.3, with an additional noise of power $g_{r2}^2 P_2$ at the relay, with a relay power $\bar{\gamma} P_R$ for the transmission to user 2 and an additional noise of power $g_{2r}^2 \gamma P_R$ at receiver 2. (The quantization index is treated as additional noise for the decoding at receiver 2.)
For user 2, we use the CF scheme presented in Section 4.1.2, with a relay power γP_R for the transmission to user 1.

□

5.2.6 Comparison of the presented protocols

In this subsection we compare the achievable rates for the full-duplex Gaussian two-way relay channel using CDF, DF and AF in a free space path loss model. We thus assume that the gains are inversely proportional to the distance between nodes raised to the power 3/2. We suppose that the source and destination are unit distance apart and that the relay is between the source and the destination at a distance d from the source: The channel gains are given as: $g_{12} = g_{21} = 1$, $g_{1r} = g_{r1} = 1/d^{3/2}$ and $g_{2r} = g_{r2} = 1/(1-d)^{3/2}$. For this numerical example, we set $P = P_R = 10$ and $N_R = N_D = 1$.

Figure 5.5 gives the achievable sum rate of CDF, DF and AF as well as the cut-set bound as a function of the distance d between the relay and user 1. We can see that DF performs close to the upper bound when the relay is close to user 1 (or user 2) and that CDF gives the best performance when the relay is close to one user. (For CDF, at $d = 0.5$, the direction of CF and DF protocols are reversed.)

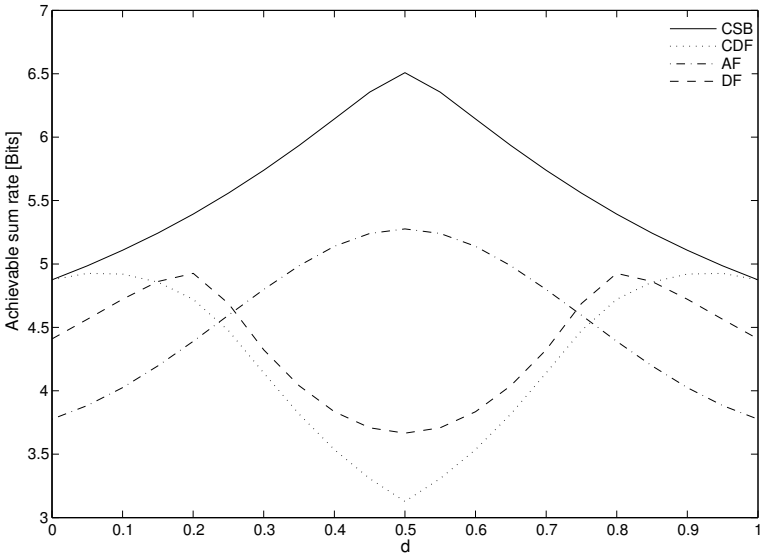


Figure 5.5: Comparison of the cut-set bound, DF, AF and CDF as the function of the distance d between user 1 and the relay ($g_{12} = g_{21} = 1$, $g_{1r} = g_{r1} = 1/d^{3/2}$, $g_{2r} = g_{r2} = 1/(1-d)^{3/2}$).

5.3 Full-duplex Gaussian two-way relay channel with correlated noises

As in the previous chapter, we investigate a more general Gaussian two-way relay channel with direct links by considering correlated additive Gaussian noises at the relay and both destinations.

As previously, in the Gaussian case, user 1 sends X_1 of power P_1, user 2 X_2 of power P_2 and the relay X_R of power P_R. The received signals are:

$$\begin{aligned} \text{At the relay: } Y_R &= g_{r1}X_1 + g_{r2}X_2 + Z_R; \\ \text{At user 1: } Y_1 &= g_{12}X_2 + g_{1r}X_R + Z_1; \\ \text{At user 2: } Y_2 &= g_{21}X_1 + g_{2r}X_R + Z_2, \end{aligned}$$

where Z_R, Z_1 and Z_2 are Gaussian noises of variance N_R, N_1, N_2 such that Z_R is correlated with

Z_1 and Z_2. The noise correlation coefficients are defined as $\rho_{z1} = \dfrac{\mathbb{E}[Z_1 Z_R]}{\sqrt{N_1 N_R}}$ and $\rho_{z2} = \dfrac{\mathbb{E}[Z_2 Z_R]}{\sqrt{N_2 N_R}}$.

5.3.1 Cut-set bound

Since the capacity of this channel remains unknown, we start by presenting an upper bound using a cut-set argument.

Proposition 5.3.1. *([Savard and Weidmann, 2015b]) For the Gaussian two-way relay channel with direct links and correlated noise, the CSB for restricted encoders is given by the convex closure of the cut-set region:*

$$\bigcup_{0\le\rho_1,\rho_2\le 1}(R_1,R_2): \left\{ R_1 \le \min\left\{ C\left(\frac{P_1(\overline{\rho_2^2}-\rho_1^2)(g_{21}^2 N_R + g_{r1}^2 N_2 - 2g_{21}g_{r1}\sqrt{N_2 N_R}\rho_{z2})}{\overline{\rho_2^2}\ \overline{\rho_{z2}^2} N_2 N_R}\right), C\left(\frac{g_{2r}^2 P_R \overline{\rho_2^2} + g_{21}^2 P_1 + 2g_{21}g_{2r}\rho_1\sqrt{P_1 P_R}}{N_2}\right)\right\}, \right.$$
$$\left. R_2 \le \min\left\{ C\left(\frac{P_2(\overline{\rho_1^2}-\rho_2^2)(g_{12}^2 N_R + g_{r2}^2 N_1 - 2g_{12}g_{r2}\sqrt{N_1 N_R}\rho_{z1})}{\overline{\rho_1^2}\ \overline{\rho_{z1}^2} N_1 N_R}\right), C\left(\frac{g_{1r}^2 P_R \overline{\rho_1^2} + g_{12}^2 P_2 + 2g_{12}g_{1r}\rho_2\sqrt{P_2 P_R}}{N_1}\right)\right\}\right\}.$$

Proof. The cut-set region is given by

$$\begin{cases} R_1 \le \min[I(X_1; Y_R, Y_2 | X_R, X_2), I(X_1, X_R; Y_2 | X_2)], \\ R_2 \le \min[I(X_2; Y_R, Y_1 | X_R, X_1), I(X_2, X_R; Y_1 | X_1)]. \end{cases}$$

We introduce two correlation coefficients $\rho_1 = \dfrac{\mathbb{E}[X_1 X_R]}{\sqrt{P_1 P_R}}$ and $\rho_2 = \dfrac{\mathbb{E}[X_2 X_R]}{\sqrt{P_2 P_R}}$.

In this proof, we only compute

$$R_1 \le \min[I(X_1; Y_R, Y_2 | X_R, X_2), I(X_1, X_R; Y_2 | X_2)]. \tag{5.6}$$

The bound on R_2 follows in analogous fashion.

The first mutual information of (5.6) is written as

$$I(X_1; Y_R, Y_2 | X_R, X_2) = H(Y_2 | X_2, X_R) + H(Y_R | Y_2, X_2, X_R) - H(Y_R, Y_2 | X_R, X_2, X_1).$$

Using the linear MMSE estimate of Y_2 given X_2 and X_R, we obtain

$$H(Y_2 | X_2, X_R) \le \frac{1}{2}\log_2\left(2\pi e \frac{g_{21}^2 P_1(\rho_1^2 - \overline{\rho_2^2}) - N_2\overline{\rho_2^2}}{-\overline{\rho_2^2}}\right).$$

Using the linear MMSE estimate of Y_R given Y_2, X_2 and X_R we obtain

$$H(Y_R | Y_2, X_2, X_R) \le \frac{1}{2}\log_2\left(2\pi e\left(N_R + \frac{N_2 N_R \rho_{z2}^2 \overline{\rho_2^2} + g_{r1}P_1(\overline{\rho_2^2} - \rho_1^2)(2g_{21}\sqrt{N_2 N_R}\rho_{z2} - g_{r1}N_2)}{-N_2\overline{\rho_2^2} + g_{21}^2 P_1(\rho_1^2 - \overline{\rho_2^2})}\right)\right).$$

$$H(Y_R, Y_2|X_R, X_2, X_1) = H(Z_R, Z_2) = \frac{1}{2}\log_2\left((2\pi e)^2|K_Z|\right),$$

where $|K_Z|$ is the determinant of the noise covariance matrix : $|K_Z| = N_2 N_R(1-\rho_{z2}^2)$.

Thus,

$$I(X_1; Y_R, Y_2|X_R, X_2) \leq C\left(\frac{P_1(\overline{\rho_2^2}-\rho_1^2)(g_{21}^2 N_R + g_{r1}^2 N_2 - 2g_{21}g_{r1}\sqrt{N_2 N_R}\rho_{z2})}{\overline{\rho_2^2}\ \overline{\rho_{z2}^2} N_2 N_R}\right).$$

The second mutual information of (5.6) is obtained as

$$I(X_1, X_R; Y_2|X_2) \leq C\left(\frac{g_{2r}^2 P_R \overline{\rho_2^2} + g_{21}^2 P_1 + 2g_{21}g_{2r}\rho_1\sqrt{P_1 P_R}}{N_2}\right).$$

□

The remainder of this section presents achievable rate regions using CDF or DF, as well as a comparison of these two protocols.

In the following, we assume that the relay is very close to user 1 (and hence far from user 2), such that it can only decode the message from user 1 but not the one of user 2. Instead of only decoding the message from user 1, the relay will also use a part of its power to send a compressed version of the message of user 2.

5.3.2 Compress/Decode-and-Forward

Proposition 5.3.2. *([Savard and Weidmann, 2015b]) For the Gaussian two-way relay channel with direct links and correlated noise, CDF with restricted encoders achieves the following rate region using lattices:*

$$\bigcup_{\substack{0\leq\alpha\leq 1\\ 0<\bar{\gamma}<1}} (R_1, R_2) : \left\{ R_1 \leq \min\left\{ C\left(\frac{g_{r1}^2\bar{\alpha}P_1}{g_{r2}^2 P_2 + N_R}\right), \frac{1}{2}\log_2\left(\frac{g_{21}^2 P_1 + g_{2r}^2 P_R + N_2 + 2g_{21}g_{2r}\sqrt{\alpha\bar{\gamma}P_1 P_R}}{g_{2r}^2\gamma P_R + N_2}\right)\right\},\right.$$

$$\left. R_2 \leq C\left(\frac{P_2}{N_1}\frac{g_{12}^2(N_R+D) + g_{r2}^2 N_1 - 2g_{12}g_{r2}\rho_{z1}\sqrt{N_1 N_R}}{N_R\overline{\rho_{z1}^2} + D}\right)\right\} \text{ with}$$

$$D = \frac{(g_{r2}^2 N_1 + g_{12}^2 N_R)P_2 + N_1 N_R\overline{\rho_{z1}^2} - 2g_{r2}g_{12}P_2\rho_{z1}\sqrt{N_1 N_R}}{g_{1r}^2\gamma P_R}.$$

At user 1, α allows to trade off power between repeating the message from the previous block and sending a new message. γ controls the power trade off at the relay between the decoded and the compressed part.

Proof. For user 1, we use the DF scheme presented in Section 4.1.3, with an additional noise of power $g_{r2}^2 P_2$ at the relay, with a relay power $\bar{\gamma}P_R$ for the transmission to user 2 and an additional noise of power $g_{2r}^2\gamma P_R$ at receiver 2. (The quantization index is treated as additional noise for the decoding at receiver 2.)

For user 2, we use the CF scheme presented in Section 4.2.2, with a relay power γP_R for the transmission to user 1.

□

5.3.3 Decode-and-Forward

Recall that, in case of correlated noise, the rate-region achieved with DF is the same as for the standard two-way relay channel, without correlated noises, since the relay decodes everything.

Proposition 5.3.3. *([Rankov and Wittneben, 2006]) For the Gaussian two-way relay channel with direct links and correlated noise, DF with restricted encoders achieves the following rate region:*

$$\bigcup_{0\leq\rho_1,\rho_2,\gamma\leq 1}(R_1,R_2):\left\{\begin{array}{l} R_1\leq \min\left\{C\left(\frac{g_{r1}^2\overline{\rho_1^2}P_1}{N_R}\right),C\left(\frac{g_{21}^2P_1+\gamma g_{2r}^2P_R+2\rho_1 g_{21}g_{2r}\sqrt{\gamma P_1P_R}}{N_2}\right)\right\},\\ R_2\leq \min\left\{C\left(\frac{g_{r2}^2\overline{\rho_2^2}P_2}{N_R}\right),C\left(\frac{g_{12}^2P_2+\overline{\gamma} g_{1r}^2P_R+2\rho_2 g_{12}g_{1r}\sqrt{\overline{\gamma}P_2P_R}}{N_1}\right)\right\},\\ R_1+R_2\leq C\left(\frac{g_{r1}^2\overline{\rho_1^2}P_1+g_{r2}^2\overline{\rho_2^2}P_2}{N_R}\right)\end{array}\right\}.$$

ρ_1 and ρ_2 allow to trade off power between repeating the messages from the previous block and sending a new ones at user 1 and 2. γ controls the power trade-off between the message intended for user 1 and 2.

5.3.4 Comparison of the presented protocols

We now compare the sum rate achieved with CDF, DF and the direct links only.

For our numerical examples, we consider the following way to assign the channel gains: $g_{r1}=g_{1r}=d$ and $g_{r2}=g_{2r}=(1-d)$ $(0\leq d\leq 1)$.

One major result of [Zhang et al., March 2011] is that if $\rho_{z1}=\frac{g_{2r}}{g_{21}}$ for the Gaussian relay channel with $N_1=N_R$, then CF can achieve the CSB.

For the two-way relay channel, numerical evaluations show that the gap between the CSB and CDF is minimal for this value of ρ_{z1}, i.e. $\rho_{z1}=\frac{(1-d)}{g_{21}}$.

On Figure 5.6 and Figure 5.7, we represent the sum rate as a function of ρ_{z1}. In both cases, the value $\rho_{z1}=\frac{(1-d)}{g_{21}}$ minimizes the gap between the CSB and CDF. In both cases, DF achieves low sum rate since the relayed links are very asymmetric and weaker than the direct links. We can note that on Figure 5.6, using only the direct links achieves a higher sum rate than using the relay, since the direct link is better than at least one of the relayed links, but on Figure 5.7 we see that when the direct link is worse than both relayed links, using the relay achieves higher sum rate. Thus, when the relayed links are weaker than the direct links, it is more efficient not to use the relay when one can choose between DF and CDF. Note that it wouldn't be neither efficient to send the message from user 1 over the direct links and only to compress the one from user 2 at the relay: in this case, the message from user 1 would act as additional noise at the relay for the compression of the message from user 2 but at the relay, the message from user 1 convey more power than the one that should be compressed, leading again to very poor performance.

5.4 Conclusion

In this chapter, we presented a detailed study of the Gaussian two-way relay channel (with and without direct links), which is a simple extension of the Gaussian relay channel.

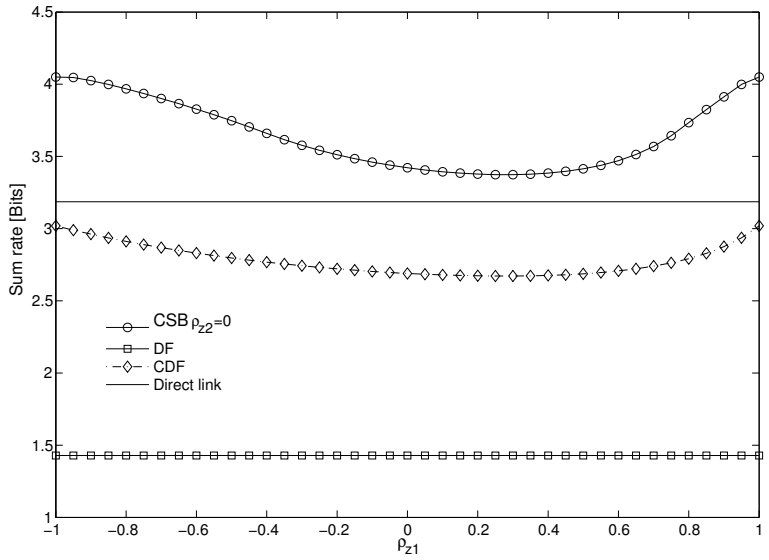


Figure 5.6: Comparison of the cut-set bound, DF, CDF and the transmission only over the direct links as the function of the noise correlation ρ_{z1}, ($g_{21} = g_{12} = 0.9$, $g_{r1} = g_{1r} = d$, $g_{r2} = g_{2r} = 1 - d$ with $d = 0.75$). Minimal gap between the cut-set bound and CDF is achieved for $\rho_{z1} = 0.27$.

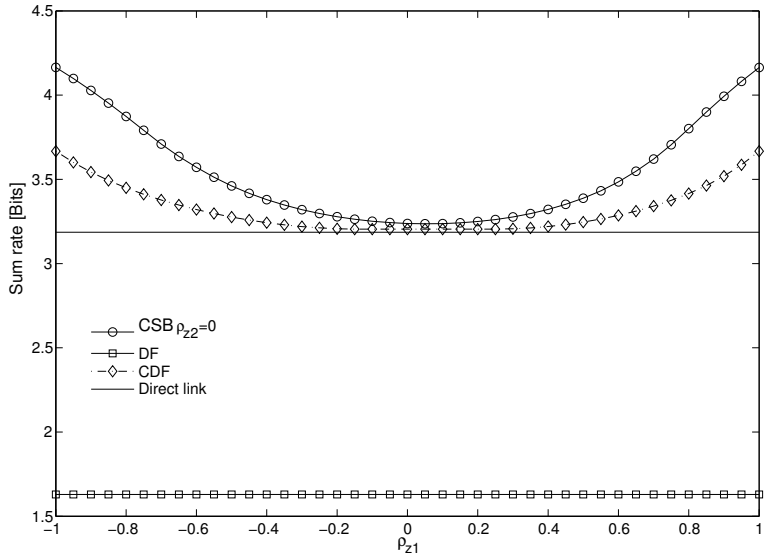


Figure 5.7: Comparison of the cut-set bound, DF, CDF and the transmission only over the direct links as the function of the noise correlation ρ_{z1}, ($g_{21} = g_{12} = 0.9$, $g_{r1} = g_{1r} = d$, $g_{r2} = g_{2r} = 1 - d$ with $d = 0.95$). Minimal gap between the cut-set bound and CDF is achieved for $\rho_{z1} = 0.05$.

We first gave an upper bound on the capacity as well as lower bounds achieved by various standard protocols such as DF, CF, AF, CDF and CoF using either lattice or AWGN coding. If the relay is between the source and the destination, we can see that if it is closer to the source, then DF outperforms CF and if the relay is closer to the destination, then CF outperforms DF.

In the second part of the chapter, we studied a more general Gaussian two-way relay channel, where the additive Gaussian noise at the relay and destinations are correlated. For this setup,

we proposed a CDF protocol combining doubly-nested lattice coding at one user and standard lattice coding at the other. Numerical examples show that, given the channel gains, CDF can outperform a transmission only over the direct links or two-way DF. We also noted that a particular value of the correlation coefficient minimizes the gap between the CDF and the CSB.

Chapter 6

The Gaussian multiway relay channel with direct links

This work was presented in [Savard and Weidmann, 2014b] and [Savard and Weidmann, 2015a].

Contents

In the previous chapter, we investigated the Gaussian two-way relay channel, a natural extension of the Gaussian relay channel, in which two users wish to exchange their messages with the help of one relay.

An extension of the TWRC named the multiway relay channel (mRC) has been recently proposed in [Gündüz et al., Jan. 2013]: the authors consider multiple clusters of users that wish to exchange messages locally within each cluster, with the help of a single relay. Achievable rates for DF, CF, AF and CoF are given for the so-called restricted model.

The main difference between the multiway relay channel in [Gündüz et al., Jan. 2013] and the model considered in this chapter consists in the inclusion of direct links between users of the same cluster. When users are close to each other (for instance in a sensor network), they can overhear signals sent by each other, thus adding direct links gives a more realistic model of the situation.

The focus of this chapter is to provide rate limits for the Gaussian mRC with intra-cluster links. The users in a cluster broadcast their messages both to the relay and to the other users within the same cluster. The relay receives incoming messages on a MAC and sends a function of its received message over a Gaussian broadcast channel to the users to help them decoding the messages they wish to receive.

We investigate two different setups in detail: the full data exchange model and the pairwise data exchange model. In the full data exchange model, each user wants to recover the messages from all users in the network. In the pairwise data exchange model, there are only two users per cluster who wish to exchange their messages. For this setup we also investigate the performance of the CoF protocol using nested lattices, which can achieve rates within half a bit from the capacity of the TWRC [Nam et al., 2008].

We focus only on one particular point in the capacity region: the *exchange rate point* for the symmetric network setup. In this setup, all users have the same power constraint and the noise powers at all terminals are the same. Moreover, the gains of the user-relay links are identical and denoted by g, and all intra-cluster links have unit gain. The *exchange rate* is the point in the capacity region with equal rate for all users. The *total exchange rate* is the total rate of all the data multicast over the network. The supremum of all achievable exchange rates is called the *exchange capacity*. We provide an upper bound on the exchange capacity and characterize the total exchange rate achievable with DF, AF, CF and CoF (when there are $K = 2$ users per cluster).

6.1 System model

This chapter considers a Gaussian multiway relay channel (mRC) in which N users, grouped into L clusters of $K \geq 2$ users each ($N = KL$), exchange messages with the help of one relay. The K users in each cluster want to recover the messages of all $K-1$ other users within their cluster. We suppose that users within a cluster are physically close, thus they can overhear the other users' messages and model this through unit gain direct links between users of the same cluster. We also assume that the relay has a better observation of the transmitted messages than the users and model this assumption through a bidirectional gain $g > 1$ on links between the relay and the users (this can be justified by better antennas and/or higher power and less noise, i.e. more powerful hardware at the relay). All nodes are assumed to be full-duplex. This model is depicted in Figure 6.1.

For the Gaussian case, user k of cluster l sends $X_{l,k}$, which is of power $P_{l,k}$ and the relay sends X_R, which is of power P_R. The received signals are:

$$\text{At user } k \text{ of cluster } l\text{: } Y_{l,k} = \sum_{\substack{i=1 \\ i \neq k}}^{K} X_{l,i} + gX_R + Z_{l,k} \tag{6.1}$$

$$\text{At the relay: } Y_R = \sum_{l=1}^{L} \sum_{k=1}^{K} gX_{l,k} + Z_R, \tag{6.2}$$

where Z_R and $Z_{l,k}$ are zero-mean, unit-variance Gaussian noises that are independent of each other and of the channel inputs. The difference with the model in [Gündüz et al., Jan. 2013] is the presence of the intra-cluster signals in (6.1).

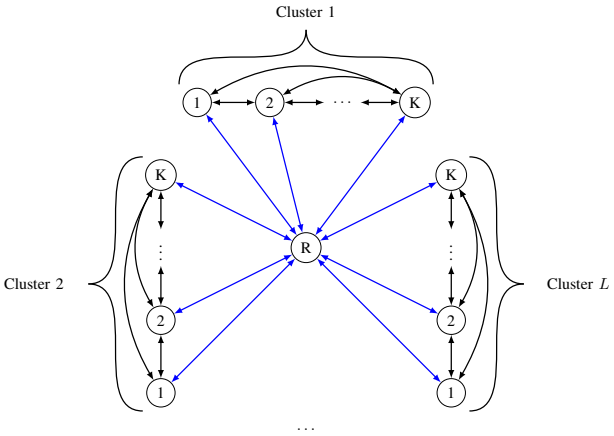


Figure 6.1: Model setup: L clusters of K users each, fully connected within a cluster, communicating over one relay; user-relay links (blue) have gain g.

6.2 Exchange rate for the symmetric network

Like Gündüz et al., we focus on a symmetric setup with equal power constraints, respectively noise variances, at the users, i.e., $P_{l,k} = P$, $\forall l, k$ and $N_{l,k} = N = 1$, $\forall l, k$. We will investigate the achievable equal rate point for different relaying schemes, i.e. $R_{l,k} = R$, $\forall l, k$.

Definition 6.2.1. *An exchange rate R_e is achievable for the mRC with L clusters of K users each if $(\frac{R_e}{KL} \dots \frac{R_e}{KL})$ is an achievable rate tuple. The exchange capacity is defined as the supremum of all achievable exchange rates.*

We also propose an upper bound on the cut-set bound, which allows an easier study of the gap between the different proposed schemes and the upper bound. Throughout this chapter, we focus on the full data exchange model. Like Gündüz et al., we first restrict analysis to time-sharing among clusters, but in Section 6.5, we relax that restriction. Since we consider a time-sharing assumption, each cluster only transmits over a $1/L$ fraction of the time, which allows us to increase the power of each user up to $P' = PL$ and still satisfy the average user power constraint.

The model (6.1) and (6.2) becomes:

$$Y_i = \sum_{\substack{k=1 \\ k \neq i}}^{K} X_k + g X_R + Z_i,$$

$$Y_R = g \sum_{k=1}^{K} X_k + Z_R,$$

where we drop the cluster index for notation simplicity.

The remainder of the section presents the cut-set bound as well as achievable rates using CF, AF, DF and CoF (only when $K = 2$). For each lower/upper bound, we also study the limit when g grows large, i.e., when the direct links become negligible compared to the relayed links. This study allows a comparison with the rates obtained by Gündüz et al.. In order to make a fair comparison, we must normalize the transmitted powers (at the users and the relay) by g^2.

6.2.1 Cut-set bound

Since the capacity of this channel remains unknown, we first give an upper bound by considering a cut-set argument.

Proposition 6.2.1. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of K users each, the CSB on the exchange capacity, for restricted encoders, is given by:*

$$R_{CSB} = \max_{\rho\in[0,1]} \frac{K}{K-1} \min\left\{f_1(\rho), f_2(\rho)\right\}, \text{ where}$$

$$f_1(\rho) = C\left(\frac{(g^2+1)P'(K-1)\left(\overline{\rho^2}-(K-1)\rho^2\right)}{\overline{\rho^2}}\right) \tag{6.3}$$

$$f_2(\rho) = C\Big((K-1)P'+g^2P_R\overline{\rho^2}+2g\sqrt{P'P_R}(K-1)\rho\Big), \tag{6.4}$$

and ρ is a correlation parameter.

Proof. Arguing that the most restricting cut is the one with all nodes but one in a set, we obtain

$$\begin{cases}(K-1)R_K \le I(X_1,\dots,X_{K-1};Y_K,Y_R|X_K,X_R)\\ (K-1)R_K \le I(X_1,\dots,X_{K-1},X_R;Y_K|X_K).\end{cases} \tag{6.5}$$

Using the linear MMSE estimate of Y_K given X_K, the second constraint in (6.5) can be upper-bounded as

$$I(X_1,\dots,X_{K-1},X_R;Y_K|X_K) \le C\left((K-1)P'+g^2P_R\overline{\rho_K^2}+2g\sqrt{P'P_R}\sum_{k=1}^{K-1}\rho_k\right), \tag{6.6}$$

where ρ_k stands for the correlation coefficient $\rho_k = \dfrac{\mathbb{E}[X_kX_R]}{\sqrt{P'P_R}}$.

We write the right-hand side of the first constraint in (6.5) as

$$\begin{aligned}I(X_1,\dots,X_{K-1};Y_K,Y_R|X_K,X_R) =& H(Y_K|X_K,X_R) + H(Y_R|X_K,X_R,Y_K)\\ &- H(Y_K,Y_R|X_1,\dots,X_K,X_R).\end{aligned}$$

Using the linear MMSE estimate of Y_K given X_K, X_R, we obtain

$$H(Y_K|X_K,X_R) \le \frac{1}{2}\log_2\left(2\pi e\left(1+\frac{P'}{\overline{\rho_K^2}}\left((K-1)\overline{\rho_K^2}-\left(\sum_{k=1}^{K-1}\rho_k\right)^2\right)\right)\right).$$

Using the linear MMSE estimate of Y_R given X_K, X_R, Y_K, we obtain

$$H(Y_R|X_K,X_R,Y_K) \le \frac{1}{2}\log_2\left(2\pi e\frac{N}{D}\right),$$

where $D = P'P_R\left(-\overline{\rho_K^2}+P'\left(-(K-1)\overline{\rho_K^2}+\left(\sum_{k=1}^{K-1}\rho_k\right)^2\right)\right)$ and $N = (g^2+1)D+g^2P'P_R\overline{\rho_K^2}$.

Thus,

$$I(X_1,\ldots,X_{K-1};Y_K,Y_R|X_K,X_R) \le \frac{1}{2}\log_2\left(\frac{(g^2+1)D+g^2P'P_R\overline{\rho_K^2}}{D}\frac{D}{-P'P_R\overline{\rho_K^2}}\right)$$
$$= C\left(\frac{(g^2+1)P'}{\overline{\rho_K^2}}\left((K-1)\overline{\rho_K^2}-\left(\sum_{k=1}^{K-1}\rho_k\right)^2\right)\right). \qquad (6.7)$$

Inequalities (6.6) and (6.7) lead to:

$$R_K \le \frac{1}{K-1}\min\left\{C\left(\frac{(g^2+1)P'}{\overline{\rho_K^2}}\left((K-1)\overline{\rho_K^2}-\left(\sum_{k=1}^{K-1}\rho_k\right)^2\right)\right),\right.$$
$$\left. C\left((K-1)P'+g^2P_R\overline{\rho_K^2}+2g\sqrt{P'P_R}\sum_{k=1}^{K-1}\rho_k\right)\right\}.$$

Since every node must satisfy the same constraints that are based on its own correlation with the relay transmitter signal and the correlation of all other signals with the relay, the above inequalities reduce to

$$R \le \max_{\rho\in[0,1]}\frac{1}{K-1}\min\Big\{f_1(\rho),f_2(\rho)\Big\}, \text{ where}$$

$$f_1(\rho) = C\left(\frac{(g^2+1)P'(K-1)\left(\overline{\rho^2}-(K-1)\rho^2\right)}{\overline{\rho^2}}\right) \text{ and}$$
$$f_2(\rho) = C\Big((K-1)P'+g^2P_R\overline{\rho^2}+2g\sqrt{P'P_R}(K-1)\rho\Big).$$

Here ρ stands for the correlation coefficient $\rho = \dfrac{\mathbb{E}[X_iX_R]}{\sqrt{P'P_R}}$, $\forall i$. □

Proposition 6.2.2. *For a symmetric Gaussian mRC with direct links, L clusters of K users each, the CSB, for restricted encoders and asymptotically large gain g (with the appropriate normalization of the powers), is given by:*

$$R_{CSB}^{g\to\infty} = \frac{K}{K-1}\min\Big\{C\big((K-1)P'\big),C(P_R)\Big\}.$$

The CSB for Gündüz et al.'s mRC model, without direct links and with relay-user links of gain 1, is given by:

$$R_{CSB}^{\text{Gündüz et al.}} = \frac{K}{K-1}\min\Big\{C\big((K-1)P'\big),C(P_R)\Big\}.$$

Remark: Note that $R_{CSB}^{g\to\infty} = R_{CSB}^{\text{Gündüz et al.}}$.

Proof. By replacing P' by P'/g^2 and P_R by P_R/g^2 in (6.3) and (6.4) and by taking the limit, we obtain

$$\lim_{g\to\infty} f_1(\rho) = C\left(\frac{(K-1)P'\left(\overline{\rho^2}-(K-1)\rho^2\right)}{\overline{\rho^2}}\right) \text{ and } \lim_{g\to\infty} f_2(\rho) = C\left(P_R\overline{\rho^2}\right).$$

We see that both limits are decreasing in ρ, thus the optimum value is $\rho = 0$, which completes the proof. □

In the next subsections, we present the exchange rate for the Amplify-and-Forward (AF), the Compress-and-Forward (CF), the Decode-and-Forward (DF) and the Compute-and-Forward (CoF) protocols.

6.2.2 Amplify-and-Forward

This part is inspired by [Chang et al., 2010], where an AF scheme has been proposed for the Gaussian relay channel. Here we extend this scheme to many users grouped into clusters.

Proposition 6.2.3. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of K users each, the exchange rate achievable with AF relaying and restricted encoders is:*

$$R_{AF} = \frac{K}{2(K-1)}\log_2\left(\frac{\alpha+\sqrt{\alpha^2-\beta^2}}{2}\right),$$

with

$$\alpha = 1+(K-1)P'\frac{g^2(KP'+g^2P_R)+1}{g^2(KP'+P_R)+1} \tag{6.8}$$

$$\beta = 2(K-1)P'g^2\frac{\sqrt{P_R(g^2KP'+1)}}{g^2(KP'+P_R)+1}. \tag{6.9}$$

Proof. The proof follows along the line of the proof of Proposition 4.1.4 using a unit-memory intersymbol MAC of $K-1$ users. The scaling factor at the relay is here $\sqrt{\frac{P_R}{g^2KP'+1}}$. □

Proposition 6.2.4. *For a symmetric Gaussian mRC with direct links, L clusters of K users, the following exchange rate is achievable with AF relaying for restricted encoders and asymptotically large gain g (with the appropriate normalization of the powers):*

$$R_{AF}^{g\to\infty} = \frac{K}{K-1}C\left(\frac{(K-1)P'P_R}{KP'+P_R+1}\right).$$

The achievable exchange rate with Gündüz et al.'s mRC model, without direct links and with relay-user links of gain 1, and AF relaying is given by:

$$R_{AF}^{\text{Gündüz et al.}} = \frac{K}{K-1}C\left(\frac{(K-1)P'P_R}{KP'+P_R+1}\right).$$

Remark: Note that $R_{AF}^{g\to\infty} = R_{AF}^{\text{Gündüz et al.}}$.

Proof. By replacing P' by P'/g^2 and P_R by P_R/g^2 in (6.8) and (6.9) and by taking the limit, we obtain $\lim_{g\to\infty} \alpha = 1 + \frac{(K-1)P'P_R}{1+P_R+KP'}$ and it's straightforward that $\lim_{g\to\infty} \beta = 0$.

Thus,

$$\lim_{g\to\infty} \frac{\alpha+\sqrt{\alpha^2-\beta^2}}{2} = \lim_{g\to\infty} \alpha = 1+\frac{(K-1)P'P_R}{1+P_R+KP'},$$

which concludes the proof. □

6.2.3 Compress-and-Forward

This part is inspired by [Song and Devroye, 2011], where a lattice-based CF scheme has been proposed for the Gaussian relay channel. Here we extend the proposed scheme to multiple nodes in a cluster.

Proposition 6.2.5. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of K users each, the exchange rate achievable with CF relaying using lattice coding and restricted encoders is:*

$$R_{CF} = \frac{K}{K-1} C\left((K-1)P'\left(1+\frac{g^2}{1+D}\right)\right) \text{ with } D = \frac{(1+g^2)(K-1)P'+1}{g^2 P_R}. \tag{6.10}$$

Proof. The proof follows along the lines of the proof of Proposition 4.1.2. The main difference are the second moment of the shaping lattice used for the quantization, $\sigma^2(\Lambda_Q) = 1+D+\frac{g^2(K-1)P'}{1+(K-1)P'}$ and the use of a $K-1$ user MAC to recover all messages. □

Proposition 6.2.6. *For a symmetric Gaussian mRC with direct links, L clusters of K users each, the following exchange rate is achievable with CF relaying using lattice codes for restricted encoders and asymptotically large gain g (with the appropriate normalization of the powers):*

$$R_{CF}^{g\to\infty} = \frac{K}{K-1} C\left(\frac{(K-1)P'P_R}{1+(K-1)P'+P_R}\right).$$

The achievable exchange rate with Gündüz et al.'s mRC model, without direct links and with relay-user links of gain 1, and CF relaying is given by:

$$R_{CF}^{\text{Gündüz et al.}} = \frac{K}{K-1} C\left(\frac{(K-1)P'P_R}{1+(K-1)P'+P_R}\right).$$

Remark: Note that $R_{CF}^{g\to\infty} = R_{CF}^{\text{Gündüz et al.}}$.

Proof. Replacing P' by P'/g^2 and P_R by P_R/g^2 in (6.10) yields

$$R_{CF} = \frac{K}{K-1} C\left(\frac{(K-1)\frac{P'}{g^2}\left(P_R+\frac{1+g^2}{g^2}P'+1+g^2P_R\right)}{P_R+\frac{1+g^2}{g^2}(K-1)P'+1}\right)$$

and

$$\lim_{g\to\infty} \frac{(K-1)\frac{P'}{g^2}\left(P_R+\frac{1+g^2}{g^2}P'+1+g^2P_R\right)}{P_R+\frac{1+g^2}{g^2}(K-1)P'+1} = \frac{(K-1)P'P_R}{1+(K-1)P'+P_R},$$

which concludes the proof. □

6.2.4 Decode-and-Forward

This part is inspired by [Khina et al., 2012], where a DF scheme using AWGN superposition coding and decoding has been proposed for the Gaussian relay channel. Here we extend this superposition scheme to multiple users in a cluster.

Proposition 6.2.7. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of K users each, the exchange rate achievable with DF relaying and restricted encoders is:*

$$R_{DF} = \max_{\rho\in[0,1]} \min\left\{R_1(\rho), R_2(\rho)\right\},$$

where

$$R_1(\rho) = C\Big(g^2\overline{\rho^2}KP'\Big) \tag{6.11}$$

$$R_2(\rho) = \frac{K}{K-1}C\left((K-1)\left(P'+g^2\frac{P_R}{K}+2g\rho\sqrt{\frac{P'P_R}{K}}\right)\right). \tag{6.12}$$

Proof. The proof follows along the lines of the proof of Proposition 5.2.2. The main differences are the powers of each part of the codeword, here $X_{k1},\ \forall k \in \{1,\dots,K\}$ is of power P_R/K, $X_{k2},\ \forall k \in \{1,\dots,K\}$ is of power P', and the use of a $K-1$ user MAC to recover all messages. □

Proposition 6.2.8. *For a symmetric Gaussian mRC with direct links, L clusters of K users each, the following exchange rate is achievable with DF relaying for restricted encoders and asymptotically large gain g (with the appropriate normalization of the powers):*

$$R_{DF}^{g\to\infty} = \min\left\{C\big(KP'\big), \frac{K}{K-1}C\left(\frac{(K-1)P_R}{K}\right)\right\}.$$

The achievable exchange rate with Gündüz et al.'s mRC model, without direct links and with relay-user links of gain 1, and DF relaying is given by:

$$R_{DF}^{\text{Gündüz et al.}} = \min\left\{C\big(KP'\big), \frac{K}{K-1}C(P_R)\right\}.$$

Remark: Note that the two achievable rates $R_{DF}^{g\to\infty}$ and $R_{DF}^{\text{Gündüz et al.}}$ differ only by the factor $\frac{K-1}{K}$ in the second term of the min function. Nevertheless, this term becomes negligible when the number of users becomes large.

Proof. By replacing P' by P'/g^2 and P_R by P_R/g^2 in (6.11) and (6.12) and by taking the limit, we obtain

$$\lim_{g\to\infty} R_1(\rho) = C(\overline{\rho^2}KP') \quad \text{and}$$
$$\lim_{g\to\infty} R_2(\rho) = \frac{K}{K-1}C\left(\frac{(K-1)P_R}{K}\right).$$

Since $R_1(\rho)$ is a decreasing function and $R_2(\rho)$ is constant in ρ, the optimum value of ρ is $\rho^* = 0$, which completes the proof. □

6.2.5 Compute-and-Forward

For this subsection, we focus on the pairwise data exchange model: there are only two users in each cluster. This part is based on [Song and Devroye, Aug. 2013], where a combination of CoF and DF has been proposed for the TWRC with unitary links between the relay and the two destinations nodes.

The main advantage of CoF is to compute directly the sum of the messages at the relay instead of decoding both messages individually.

Proposition 6.2.9. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of $K = 2$ users each, the exchange rate achievable with CoF relaying and restricted encoders is:*

$$R_{CoF} = \min\left\{\log_2^+\left(\frac{1}{2}+g^2P'\right), \log_2(1+P'+g^2P_R)\right\}. \tag{6.13}$$

Proof. The following nested lattices are used: $\Lambda \subseteq \Lambda_s \subseteq \Lambda_c$, $\Lambda_R \subseteq \Lambda_{sR} \subseteq \Lambda_{cR}$, with $\sigma^2(\Lambda) = P'$ and $\sigma^2(\Lambda_R) = P_R$.

Transmitters send:

$$X_i(b) = [t_i(b)+U_i(b)] \mod \Lambda,\ i = 1, 2$$

where U_i is a dither uniformly distributed over $\mathcal{V}$.

The relay receives $Y_R = g(X_1+X_2)+Z_R$ and can compute $T = [t_1+t_2] \mod \Lambda$ if

$$R \leq \frac{1}{2}\log_2^+\left(\frac{1}{2}+g^2P'\right).$$

Then, using the list decoder proposed in [Song and Devroye, Aug. 2013], the destinations can decode each other's messages as long as

$$R \leq \frac{1}{2}\log_2(1+P'+g^2P_R).$$

□

Proposition 6.2.10. *For a symmetric Gaussian mRC with direct links, L clusters of $K = 2$ users each, the following exchange rate is achievable with CoF relaying using lattice codes, for restricted encoders and asymptotically large gain g (with the appropriate normalization of the powers):*

$$R_{CoF}^{g\to\infty} = \min\left\{\log_2^+\left(\frac{1}{2}+P'\right), \log_2(1+P_R)\right\}.$$

The achievable exchange rate with Gündüz et al.'s mRC model, without direct links and with relay-user links of gain 1, and CoF relaying is given by:

$$R_{CoF}^{\text{Gündüz et al.}} = \min\left\{\log_2^+\left(\frac{1}{2}+P'\right), \log_2(1+P_R)\right\}.$$

Remark: Note that $R_{CoF}^{g\to\infty} = R_{CoF}^{\text{Gündüz et al.}}$.

Proof. By replacing P' by P'/g^2 and P_R by P_R/g^2 in (6.13), and by taking the limit, the result is straightforward. □

6.3 Comparison with the cut-set bound

In this section, we characterize the gaps to the cut-set bound of our proposed schemes. In particular, we prove that the proposed schemes can achieve a finite gap that is independent of the transmit powers and number of clusters of the system. We also prove that the AF protocol achieves a finite gap to the CF protocol.

We first derive an upper bound on the cut-set bound which will be useful to analyze the performance of the proposed protocols. The goal is to obtain a bound based only on the system parameters and not on the optimization parameter ρ corresponding to the correlation coefficient between the relay and the users.

6.3.1 Weakening the cut-set bound

Proposition 6.3.1. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of K users each, the CSB for restricted encoders can be upper bounded by:*

$$R_{CSB} \leq \frac{K}{K-1} C\big((g^2+1)(K-1)P'\big)\,. \tag{6.14}$$

Proof. Recall that

$$R_{CSB} = \max_{\rho\in[0,1]} \frac{K}{K-1} \min\left\{f_1(\rho), f_2(\rho)\right\}, \text{ where}$$

$$f_1(\rho) = C\left(\frac{(g^2+1)P'(K-1)\left(\overline{\rho^2}-(K-1)\rho^2\right)}{\overline{\rho^2}}\right) \text{ and}$$
$$f_2(\rho) = C\Big((K-1)P'+g^2P_R\overline{\rho^2}+2g\sqrt{P'P_R}(K-1)\rho\Big)\,.$$

First, it can be proven that $f_1(\rho)$ is a decreasing function such that

$$\forall\rho\in[0,1], \quad f_1(\rho) \leq \frac{1}{2}C\big((g^2+1)(K-1)P'\big)$$

and that $f_2(\rho)$ is a concave function whose maximum $C\big((K-1)P'+g^2P_R+(K-1)^2P'\big)$ is reached for $\rho^* = \frac{K-1}{g}\sqrt{\frac{P'}{P_R}}$.

The final result relies on the study of the relative position of $f_1(\rho)$ and $f_2(\rho)$. We can notice that $f_1(1) \leq f_2(1)$. Based on the relative position of $f_1(0)$ and $f_2(0)$, and on the increasing or concave nature of $f_2(\rho)$, there are either zero, or more intersections points , but in each case, the maximum value that R_{CSB} can take is upper-bounded by the value $f_1(0) = C\big((g^2+1)(K-1)P'\big)$. □

6.3.2 Gaps to cut-set bound and comparison between schemes

Proposition 6.3.2. *([Savard and Weidmann, 2014b]) For a symmetric Gaussian mRC with direct links, L clusters of K users each and restricted encoders, the CF protocol achieves rates within $\frac{K}{2(K-1)}\log_2(1+g^2)$ bits of the exchange capacity.*

Proof.

$$R_{CF} = \frac{K}{2(K-1)} \log_2 \left((1+(g^2+1)(K-1)P')\right) - \frac{K}{2(K-1)} \log_2 \left(\frac{1+(1+g^2)(K-1)P'+g^2P_R}{1+(K-1)P'+g^2P_R} \right)$$
$$\geq R_{CSB} - \frac{K}{2(K-1)} \log_2 \left(\frac{1+(1+g^2)(K-1)P'+g^2P_R}{1+(K-1)P'+g^2P_R} \right)$$
$$\geq R_{CSB} - \frac{K}{2(K-1)} \log_2(1+g^2).$$

Justification of the two last steps:
Define the function $f_1(P_R) = \frac{1+(1+g^2)(K-1)P'+g^2P_R}{1+(K-1)P'+g^2P_R}$. It can be shown that $f_1(P_R)$ is a strictly decreasing function and that

$$\forall P_R,\ f_1(P_R) \leq f_2(P') = \frac{1+(1+g^2)(K-1)P'}{1+(K-1)P'}.$$

Then, it can be proven that $f_2(P')$ is an increasing function and that

$$\forall P',\ f_2(P') \leq 1+g^2,$$

which completes the proof. □

Proposition 6.3.3. *For a symmetric Gaussian mRC with direct links, L clusters of K users each and restricted encoders, the AF protocol achieves rates within $\frac{K}{2(K-1)} \log_2(2(g^2+1))$ bits of the exchange capacity.*

Proof. Recall that

$$R_{AF} = \frac{K}{2(K-1)} \log_2 \left(\frac{\alpha+\sqrt{\alpha^2-\beta^2}}{2} \right),$$

with

$$\alpha = 1+(K-1)P' \frac{g^2(KP'+g^2P_R)+1}{g^2(KP'+P_R)+1} \text{ and } \beta = 2(K-1)P'g^2 \frac{\sqrt{P_R(g^2KP'+1)}}{g^2(KP'+P_R)+1}.$$

Define the function $f(P_R) = \alpha^2 - \beta^2$. It can be proven that $f(P_R)$ is convex and that

$$\forall P_R,\ \frac{4g^2\big(P'(g^2+1)(K-1)+1\big)}{(g^2+1)^2} \leq f(P_R) \leq (1+g^2(K-1)P')^2. \tag{6.15}$$

Let us now study α as a function of P_R. We can prove that $\alpha(P_R)$ is an increasing function and that

$$\forall P_R,\ 1+(K-1)P' \leq \alpha(P_R) \leq 1+g^2(K-1)P'. \tag{6.16}$$

Define D as $D = (1+(K-1)P')(g^2+1)+2g\sqrt{P'(g^2+1)(K-1)+1}$.
Thus, using (6.15) and (6.16) we obtain

$$-R_{AF} \leq \frac{K}{2(K-1)} \log_2 \left(\frac{2(g^2+1)}{D} \right), \text{ and}$$
$$R_{AF} \geq R_{CSB} - \frac{K}{2(K-1)} \log_2 \left(\frac{2(g^2+1)(1+(g^2+1)(K-1)P')}{D} \right)$$
$$R_{AF} \geq R_{CSB} - \frac{K}{2(K-1)} \log_2(2(g^2+1)).$$

Justification of the last step:
Define $f(P')$ as $f(P') = \frac{2(g^2+1)(1+(g^2+1)(K-1)P')}{D}$.
This function $f(P')$ is strictly increasing and $\forall P',\ f(P') \leq 2(g^2+1)$, which completes the proof. □

Proposition 6.3.4. *For a symmetric Gaussian mRC with direct links, L clusters of K users each and restricted encoders, the AF protocol achieves rates within $\frac{K}{2(K-1)} \log_2(2(1+g^2))$ bits of the CF protocol.*

Proof. We first study the achievable rate of CF as a function of P_R. It can be shown that it is an increasing function upper bounded by $\frac{1}{2(K-1)} \log_2\left(1+(g^2+1)(K-1)P'\right)$.
Thus, we have

$$R_{CF} - R_{AF} \leq \frac{1}{2(K-1)} \log_2\left(2(g^2+1)\right).$$

Notice that this is the exact same computation as for the comparison between the CSB and AF. □

6.4 Numerical results

In Figure 6.2, we plot the cut-set bound, the achievable symmetric rate for the mRC with $L = 1$ cluster of $K = 2$ and $K = 20$ users as a function of P' when $P_R = KP'$. We can see that the gap between the cut-set bound and DF diverges quickly with increasing power and this especially for a small number of users per cluster. We can also notice the finite gap between the cut-set bound and the CF protocol at all power levels, and see that AF follows CF with a constant gap. For a small number of users, CF dominates DF. Similar results are obtained when the relay doesn't

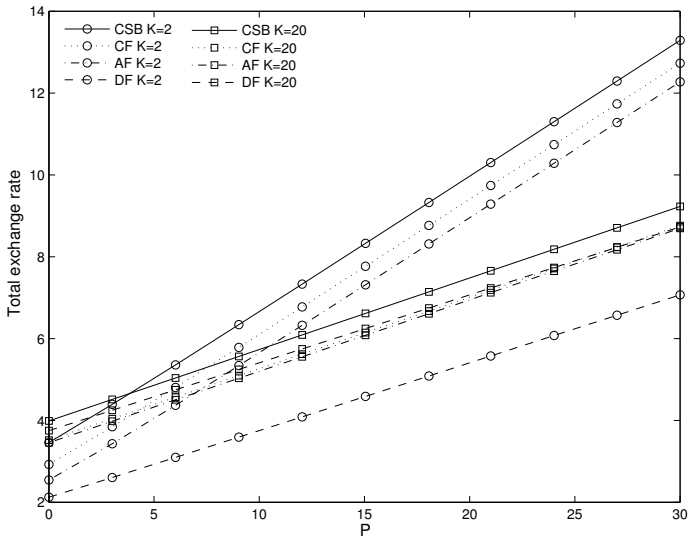


Figure 6.2: Total exchange rate in bits vs. P, $P_R = KP$, $g = 3$, $L = 1$

scale its transmit power with the number of users ($P_R = P'$), which is depicted on Figure 6.3

In Figure 6.4, we plot the cut-set bound, the achievable symmetric rate for the mRC with $L = 1$ cluster and $P' = 30dB$ as a function of K. We consider two cases: in the first one, the relay scales its power with the number of users (curves with circles) and in the second one it

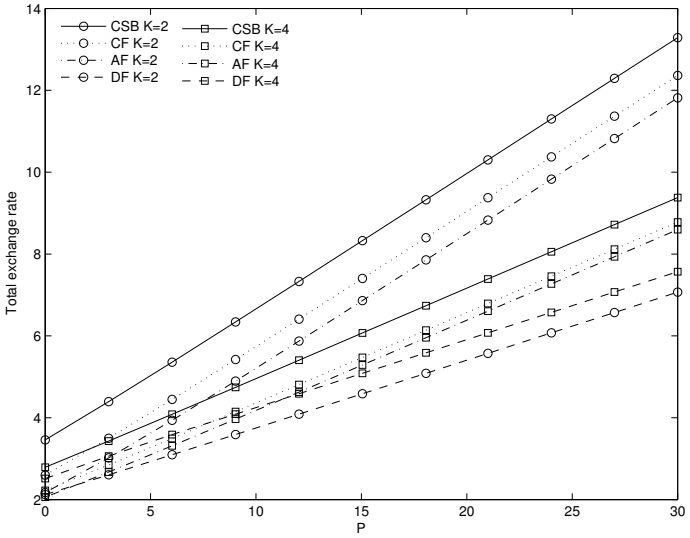


Figure 6.3: Total exchange rate in bits vs. P, $P_R = P$, $g = 3$, $L = 1$

doesn't (curves with squares). We observe that DF achieves the cut-set bound when the relay power doesn't scale with the number of users. We can also notice that in both cases, AF and CF have very close performances.

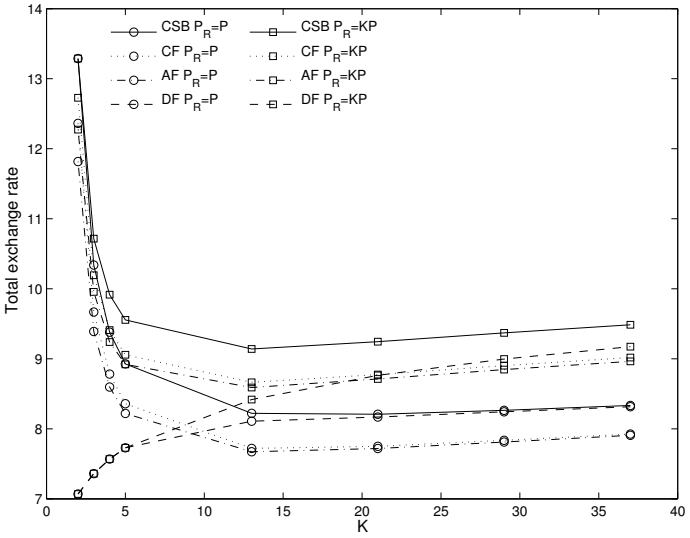


Figure 6.4: Total exchange rate in bits vs. K, $P = 30dB$, $g = 3$, $L = 1$

In Figure 6.5, we plot the cut-set bound, the achievable symmetric rate for the mRC with $L = 8$ clusters of $K = 2$ users as a function of P'. We can note that for the chosen g, CoF gives the best performances among the proposed schemes. We can also see that the gap between the cut-set bound and CoF tends to zero.

6.5 Without time-sharing between clusters

In this section, we study the Gaussian mRC with direct links, when we relax assumption that clusters are operated in time-sharing fashion (as in [Gündüz et al., Jan. 2013]). In this case, the

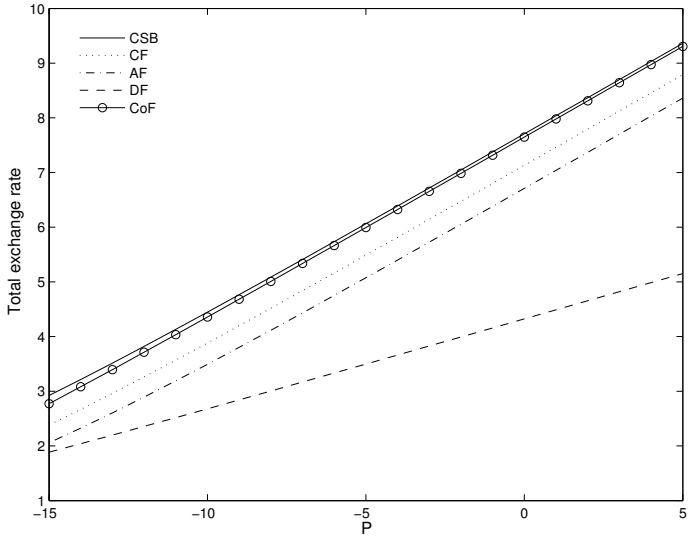


Figure 6.5: Total exchange rate in bits vs. P, $P_R = 2LP$, $g = 5$, $L = 8$

received signals are:

$$\text{At user } k \text{ of cluster } l\text{: } Y_{l,k} = \sum_{i \neq k} X_{l,i} + gX_R + Z_{l,k}$$

$$\text{At the relay: } Y_R = \sum_{l=1}^{L} \sum_{k=1}^{K} gX_{l,k} + Z_R,$$

where all users have a power constraint P and the relay a power constraint P_R.

As above, we first derive an upper bound on the capacity using a cut-set argument and then propose various lower bounds using AF, DF or CF.

6.5.1 Cut-set bound

Proposition 6.5.1. *For a symmetric Gaussian multiway relay channel with direct links, L clusters of K users each, with restricted encoders and without time-sharing, the CSB is given by:*

$$R_{CSB} \leq \max_{\rho \in [0,1]} \frac{KL}{K-1} \min\left(f_1(\rho), f_2(\rho)\right)$$

where

$$\begin{aligned}
f_1(\rho) &= C\left(\frac{N}{\overline{\rho^2}}\right) \\
f_2(\rho) &= C\left((K-1)P + g^2 P_R \overline{\rho^2} + 2g\sqrt{PP_R}(K-1)\rho\right) \\
N &= \left(Pg^2K(L-1)\right)\left(P_R(g-1)^2K(L-1)\rho^2\overline{\rho^2} - P(K-1)(KL\rho^2-1)\right) \\
&\quad - P\left(g^2(KL-1)(KL\rho^2-1) + (K-1)(K\rho^2-1)\right).
\end{aligned}$$

Proof. Arguing that the most restricting cut is the one with all nodes but one in a set, we obtain

$$\begin{cases} (K-1)R_K \leq I\left(X_{1,1},\dots,X_{1,K-1},\bigcup\limits_{k,l\neq 1} X_{l,k};Y_{1,K},Y_R|X_{1,K},X_R\right) \\ (K-1)R_K \leq I\left(X_{1,1},\dots,X_{1,K-1},\bigcup\limits_{k,l\neq 1} X_{l,k},X_R;Y_{1,K}|X_{1,K}\right). \end{cases} \tag{6.17}$$

We directly use the fact that for the exchange rate, the correlation coefficients are such that $\rho_{l,k} = \frac{\mathbb{E}[X_{l,k}X_R]}{\sqrt{PP_R}} = \rho \quad \forall l,k.$

We write the right-hand side of the first constraint in (6.17) as

$$I\left(X_{1,1},\dots,X_{1,K-1},\bigcup_{k,l\neq 1} X_{l,k};Y_{1,K},Y_R|X_{1,K},X_R\right) = H(Y_{1,K}|X_R,X_{1,K}) + H(Y_R|Y_{1,K},X_R,X_{1,K}) - H\left(Y_{1,K},Y_R|\bigcup_{k,l} X_{l,k},X_R\right)$$

Using the linear MMSE estimate of $Y_{1,K}$ given X_R and $X_{1,K}$, we obtain

$$H(Y_{1,K}|X_R,X_{1,K}) \leq \frac{1}{2}\log_2\left(2\pi e\left(1+P(K-1)\frac{\overline{\rho^2}-(K-1)\rho^2}{\overline{\rho^2}}\right)\right).$$

Using the linear MMSE estimate of Y_R given $Y_{1,K}$, X_R and $X_{1,K}$, we obtain

$$H(Y_R|Y_{1,K},X_R,X_{1,K}) \leq \frac{1}{2}\log_2\left(2\pi e\left(\frac{N+\overline{\rho^2}}{\overline{\rho^2}+P(K-1)\left(\overline{\rho^2}-(K-1)\rho^2\right)}\right)\right),$$

where

$$\begin{aligned} N = &\left(Pg^2K(L-1)\right)\left(P_R(g-1)^2K(L-1)\rho^2\overline{\rho^2}-P(K-1)(KL\rho^2-1)\right) \\ &- P\left(g^2(KL-1)(KL\rho^2-1)+(K-1)(K\rho^2-1)\right) \end{aligned}$$

Thus

$$I\left(X_{1,K},\dots,X_{1,K-1},\bigcup_{k,l\neq 1} X_{l,k};Y_{1,K},Y_R|X_{1,K},X_R\right) \leq C\left(\frac{N}{\overline{\rho^2}}\right).$$

The second constraint in (6.17) is the same as with the time-sharing assumption since the signals from other clusters do not show up in $Y_{1,K}$, thus

$$I\left(X_{1,1},\dots,X_{1,K-1},\bigcup_{k,l\neq 1} X_{l,k},X_R;Y_{1,K}|X_{1,K}\right) \leq C\left((K-1)P+g^2P_R\overline{\rho^2}+2g\sqrt{PP_R}(K-1)\rho\right).$$

□

6.5.2 Amplify-and-Forward

In the following, two versions of AF are proposed: in the first one, relayed messages from other clusters are treated as noise when recovering the messages for a given cluster, whereas in the second one, they are first decoded in order to remove them, before decoding the messages for a given cluster.

Proposition 6.5.2. *For a symmetric Gaussian multiway relay channel with direct links, L clusters of K users each, for restricted encoders and without time-sharing (relayed messages from users in other clusters are treated as noise), the following exchange rate is achievable with AF relaying:*

$$R_{AF} = \frac{KL}{2(K-1)} \log_2 \left(\frac{\alpha + \sqrt{\alpha^2 - \beta^2}}{2} \right),$$

with

$$\alpha = 1 + (K-1)P \frac{g^2(KLP + g^2 P_R) + 1}{g^2(KLP + P_R + g^2(L-1)KPP_R) + 1}$$
$$\beta = 2(K-1)Pg^2 \frac{\sqrt{P_R(g^2 KLP + 1)}}{g^2(KLP + P_R + g^2(L-1)KPP_R) + 1}.$$

Proof. The proof follows along the lines of the proof of Proposition 4.1.4 using a unit-memory intersymbol MAC of $K-1$ users. The scaling factor at the relay equals $\sqrt{\frac{P_R}{g^2 KLP+1}}$. Relayed messages from users in other clusters are treated as noise when decoding the $K-1$ messages of a given cluster. □

Proposition 6.5.3. *For a symmetric Gaussian multiway relay channel with direct links, L clusters of K users each, for restricted encoders and without time-sharing (messages from users in another cluster are decoded first to reduce the noise at each user), the following exchange rate is achievable with AF relaying:*

$$R_{AF} = \min \left\{ \frac{L}{L-1} C \left(\frac{g^4 K(L-1) P P_R}{1 + g^2(KLP + P_R) + (K-1)P(1 + g^2 KLP + g^4 P_R)} \right), \right.$$
$$\left. \frac{KL}{2(K-1)} \log_2 \left(\frac{\alpha + \sqrt{\alpha^2 - \beta^2}}{2} \right) \right\} \tag{6.18}$$

with

$$\alpha = 1 + (K-1)P \frac{g^2(KLP + g^2 P_R) + 1}{g^2(KLP + P_R) + 1} \text{ and } \beta = 2(K-1)Pg^2 \frac{\sqrt{P_R(g^2 KLP + 1)}}{g^2(KLP + P_R) + 1}.$$

Proof. The proof follows along the line of the proof of Proposition 4.1.4 using a unit-memory intersymbol MAC of $K-1$ users. The scaling factor at the relay equals $\sqrt{\frac{P_R}{g^2 KLP+1}}$. Relayed messages from other clusters are first decoded, yielding the first rate constraint in (6.18) and then removed. The $K-1$ messages of a given cluster are then decoded using the $K-1$ user MAC, yielding the second rate constraint in (6.18). □

6.5.3 Decode-and-Forward

In the following, two versions of DF are proposed: in the first one, relayed messages from other clusters are treated as noises when recovering the messages for a given cluster, whereas in the second one, they are first decoded in order to remove them, before decoding the messages for a given cluster.

Proposition 6.5.4. *For a symmetric Gaussian multiway relay channel with direct links, L clusters of K users each, for restricted encoders and without time-sharing (relayed messages*

from users in other clusters are treated as noise), the following exchange rate is achievable with DF relaying:

$$R_{DF} = \max_{\rho\in[0,1]} \min \left\{ C\left(g^2 LKP\overline{\rho^2}\right), \frac{KL}{K-1} C\left((K-1)\frac{P+g^2\frac{P_R}{KL}+2g\rho\sqrt{\frac{PP_R}{KL}}}{1+g^2\frac{L-1}{L}P_R}\right) \right\}.$$

Proof. The proof follows along the lines of the proof of Proposition 5.2.2. The main differences are the powers of each part of the codeword, here $X_{l,k,1},\ \forall k \in \{1,\dots,K\},\ \forall l \in \{1,\dots,L\}$ is of power $P_R/(KL)$, $X_{l,k,2},\ \forall k \in \{1,\dots,K\},\ \forall l \in \{1,\dots,L\}$ is of power P, and the use of a $K-1$ users MAC to recover all messages. Relayed messages from users in other clusters are treated as noise when decoding the $K-1$ messages of a given cluster. □

Proposition 6.5.5. *For a symmetric Gaussian multiway relay channel with direct links, L clusters of K users each, for restricted encoders and without time-sharing (relayed messages from users in other clusters are decoded first to reduce the noise at each user), the following exchange rate is achievable with DF relaying:*

$$R_{DF} = \max_{\rho\in[0,1]} \min \left\{ C\left(g^2 LKP\overline{\rho^2}\right), \frac{KL}{K-1} C\left((K-1)\overline{\rho^2}P\right) + \right.$$

$$\left. \min\left\{ \frac{L}{L-1} C\left(\frac{g^2 K(L-1)\frac{P_R}{KL}}{(K-1)(P+g^2\frac{P_R}{KL}+2g\rho\sqrt{\frac{PP_R}{KL}})+1}\right), \frac{KL}{K-1} C\left((K-1)\frac{g^2\frac{P_R}{KL}+\rho^2 P+2g\rho\sqrt{\frac{PP_R}{KL}}}{(K-1)\overline{\rho^2}P+1}\right) \right\} \right\}. \tag{6.19}$$

Proof. The proof follows along the lines of the proof of Proposition 5.2.2. The main differences are the power of each part of the codeword, here $X_{l,k,1},\ \forall k \in \{1,\dots,K\},\ \forall l \in \{1,\dots,L\}$ is of power $P_R/(KL)$, $X_{l,k,2},\ \forall k \in \{1,\dots,K\},\ \forall l \in \{1,\dots,L\}$ is of power P, and the use of a $K-1$ users MAC to recover all messages. The first rate constraint in (6.19) corresponds to the MAC constraint at the relay, where all KL codewords $X_{l,k,1}$ are decoded. The second rate constraint corresponds to the decoding of the $K-1$ codewords $X_{l,k,2}$ after all $X_{l,k,1}$ have been removed, which is possible as long as the minimum constraint of (6.19) is satisfied (the first term in the min corresponds to the decoding of all $(L-1)K$ codewords $X_{l,k,1}$ and the second one to the decoding of the $K-1$ codewords $X_{l,k,1}$ of a given cluster after having the $(L-1)K$ codewords $X_{l,k,1}$ removed. □

6.5.4 Compress-and-Forward

Proposition 6.5.6. *For a symmetric Gaussian multiway relay channel with direct links, L clusters of K users each, for restricted encoders and without time-sharing, the following exchange rate is achievable with CF relaying:*

$$R_{CF} = \frac{KL}{(K-1)} C\left(1+(K-1)P\left(1+\frac{g^2}{1+D+g^2(L-1)KP}\right)\right) \quad \textit{with}$$

$$D = \frac{1+(K-1)P(1+g^2) + g^2(L-1)KP(1+(K-1)P)}{g^2 P_R}.$$

Proof. The proof follows along the lines of the proof of Proposition 4.1.2. The main difference are the second moment of the shaping lattice used for the quantization, $\sigma^2(\Lambda_Q) = 1+D+\frac{g^2(K-1)P+g^2(L-1)KP((K-1)P+1)}{(K-1)P+1}$ and the use of a $K-1$ users MAC to recover all messages. Messages from users in another cluster are treated as noise when decoding the $K-1$ messages of a given cluster. □

6.6 Comparison of mRC with and without time-sharing

One interesting question concerning the multiway relay channel is whether it is more interesting to assume time-sharing among the clusters or not. Figure 6.6 and Figure 6.7 present numerical results obtained for $L = 10$ clusters of $K = 20$ users each and $g = 3$ and either $P_R = P$ or $P_R = KLP$.

In each case, results obtained with and without time-sharing are presented. The cut-set bound without time-sharing is much higher than with time-sharing and one can note that only CF performs clearly better without time-sharing than with this constraint, all other protocols perform either in the same order of magnitude or worse.

When the relay does not scale its power with the number of users, one can note that AF without time-sharing and considering additional signals from other clusters as additional noise can give better performance than with time-sharing for a large range of user power P.

When the relay scales its power with the number of users, we can note that DF, when additional signals are decoded first, gives results close to the one obtained with time-sharing and clearly outperforms the version where the additional signals are treated as noise. For the two versions of AF, the same observation can be made: decoding the additional signals first yields higher exchange rate.

Clearly, in both cases, only CF seems to be able to achieve rates close to the cut-set bound. All protocols have difficulties to deal with the additional signals, intended to other clusters and broadcast by the relay. The major issue is that these signals have a higher power than the signals that the users of a cluster want to recover.

6.7 Conclusions

In this chapter we considered an extension of the multiway relay channel proposed by [Gündüz et al., Jan. 2013]. In the considered setup, multiple clusters of users with direct intra-cluster links communicate with the help of a single relay. Each user wishes to recover all messages within its cluster. We extended standard schemes such as CF, DF, AF for this setup using results proposed for the Gaussian relay channel based on lattices [Nazer and Gastpar, Oct. 2011], [Song and Devroye, Aug. 2013] [Song and Devroye, 2011] or standard AWGN coding/decoding [Khina et al., 2012] [Chang et al., 2010]. We characterized the achievable exchange rate for all these protocols with or without time-sharing among the clusters. In the case in which there are only two users per cluster, we have proposed an extension of CoF. When we consider the time-sharing assumption, we also studied gaps to the cut-set bound that these protocols can achieve, and proved that the gaps only depend of the number of users K and on the weight g of the links between users and the relay, and do not depend on the transmit power nor on the number of clusters L. We also proved that AF performs within a finite gap from CF. For very

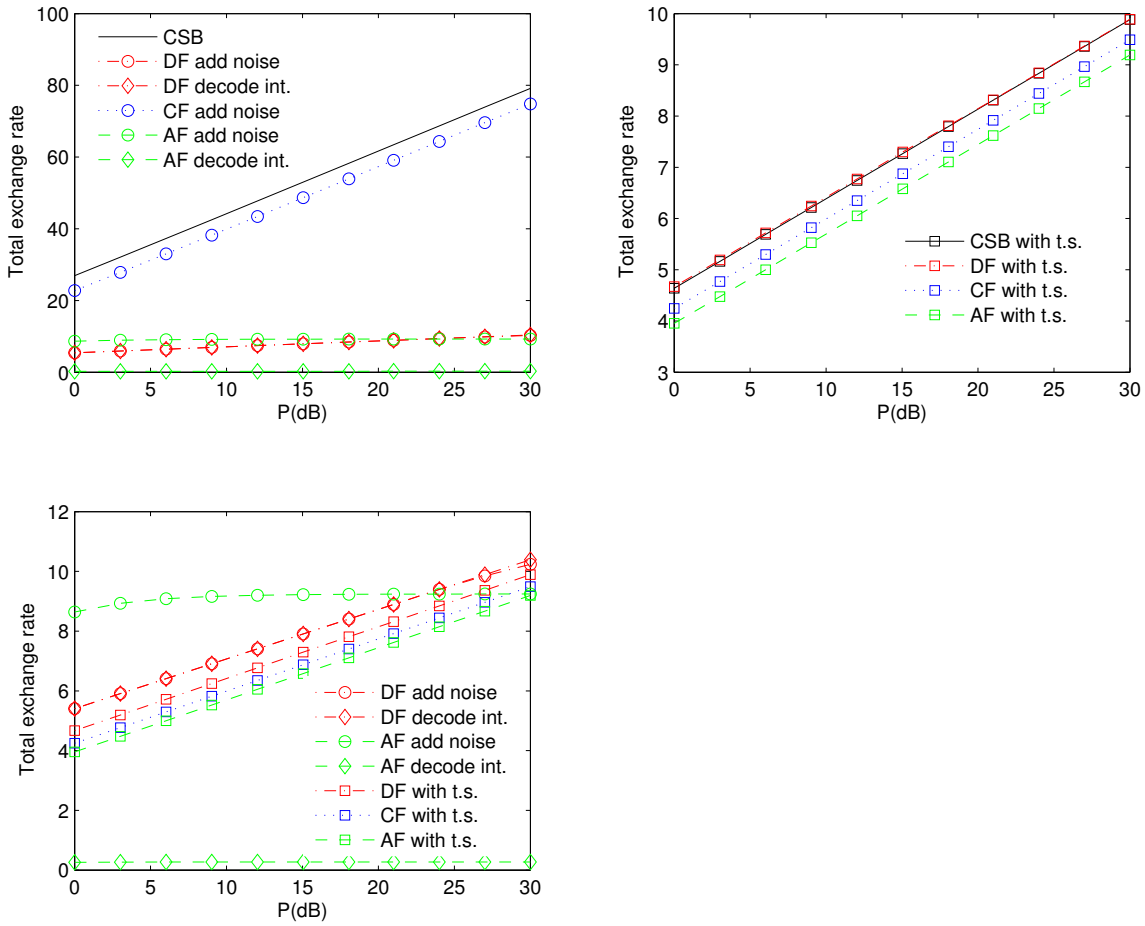


Figure 6.6: Comparison of the proposed protocols with or without time sharing among the clusters, $K = 20$, $L = 10$, $g = 3$, $P_R = P$

large user-relay gain g, i.e. when the model becomes that of [Gündüz et al., Jan. 2013] (up to scaling), the behaviors of the proposed protocols become the same as the ones obtained by [Gündüz et al., Jan. 2013] (by scaling the node and relay transmit powers accordingly). We also noted that for the general case without time-sharing, only CF performs clearly better than with the time-sharing assumption. Other protocols perform either close to the performance obtained with time-sharing or worse. This degradation is due to the interference caused by signals intended to other clusters, that are broadcast to all clusters by the relay.

Up so far, we only characterized the symmetric case. One first question that arises, is how to characterize an asymmetric case, in terms of power or gains on the links? One should find a way to compare results for this setup: a minimum achievable rate, the sum-rate... Also, we only considered a single-antenna relay: increasing the number of antennas at the relay up to K could achieve higher rates.

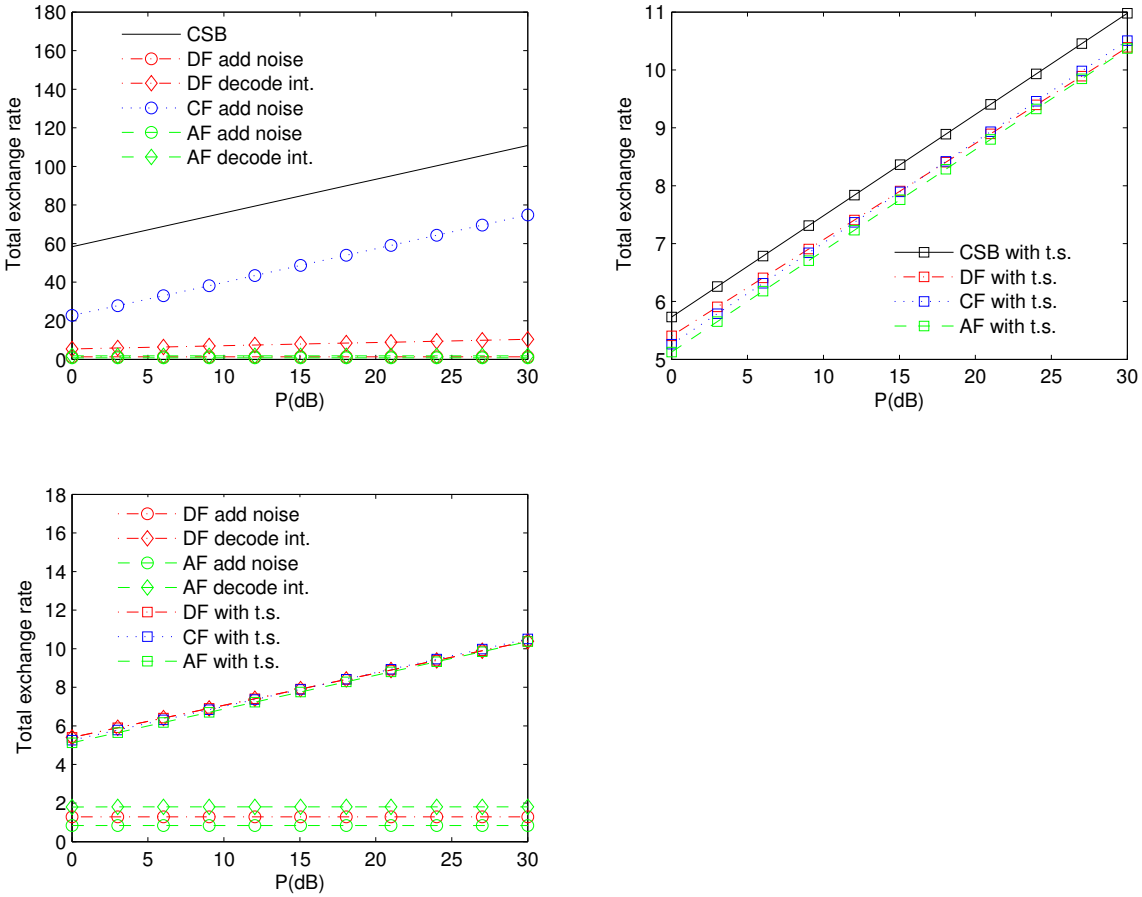


Figure 6.7: Comparison of the proposed protocols with or without time sharing among the clusters, $K = 20$, $L = 10$, $g = 3$, $P_R = KLP$

Summary and conclusion of Part II

In this part, we focused on various Gaussian full-duplex relay channel models: the relay channel, the two-way relay channel and the multiway relay channel, which is a straightforward generalization of the two-way relay channel. For the first two models, we also studied a generalization considering correlated noises, which can occur for instance when there is a common interference signal that cannot be decoded/removed.

For all these models, we first proposed a cut-set upper bound on capacity. Then, we characterized achievable lower bounds on the capacity by focusing on five state-of-the-art relaying schemes: Decode-and-Forward, Compress-and-Forward, Amplify-and-Forward, Compress/Decode-and-Forward and Compute-and-Forward.

The goal is to show that standard techniques can be applied to new channel models and generalizations of the relay channel. It is shown that lattices can achieve the theoretical rate region for CF on the relay channel with correlated noises. This result is then extended to a mixed protocol for the two-way relay channel with correlated noises. The main contribution of this second part is the extension of the multiway relay channel by adding direct links between users in a cluster.

Nowadays, almost everyone has a cell phone and many users use their communication devices at the same time, thus it is unrealistic to consider the case where there is only one user and one relay. For example in sensor network, users can be grouped into clusters that wish to exchange their data locally within the same cluster. The multiway relay channel presented here is a first attempt to characterize this situation by considering a symmetric case with direct links between users to model the overhearing of each other's communications. A more realistic setup would be the asymmetric case. Furthermore, we didn't propose any new relaying schemes, which could potentially achieve higher sum rate or could overcome the interference problem that occurs when no time-sharing is performed among clusters.

Bibliography

E. Agrell. Voronoi regions for binary linear block codes. *IEEE Transactions on Information Theory*, 42(1):310–316, Jan. 1996.

R. Ahlswede and J. Körner. Source coding with side information and a converse for degraded broadcast channels. *IEEE Transactions on Information Theory*, 21(6):629–637, Nov. 1975.

R. B. Ash. *Information Theory*. Dover, 1990.

L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2):284–287, Mar. 1974.

M. R. Best, M. V. Burnashev, Y. Lvy, A. Rabinovich, P. C. Fishburn, A. R. Calderbank, and D. J. Costello. On a technique to calculate the exact performance of a convolutional code. *IEEE Transactions on Information Theory*, 41(2):441–447, March 1995.

A. R. Calderbank, P. C. Fishburn, and A. Rabinovich. Covering properties of convolutional codes and associated lattices. *IEEE Transactions on Information Theory*, 41(3):732–746, May 1995.

W. Chang, S. Chung, and Y. H. Lee. Gaussian relay channel capacity to within a fixed number of bits. *Computing Research Repository - CORR*, 2010.

J. Conway, N. J. A. Sloane, and E. Bannai. *Sphere Packings, Lattices and Groups*. Third edition edition, 1999.

T. M. Cover and A. E. El Gamal. Capacity theorems for the relay channel. *IEEE Transactions on Information Theory*, 25(5):572–584, Sept. 1979.

T. M. Cover and J. A. Thomas. *Elements of information theory,* 2^{nd} *edition*. Wiley, 2006.

R. de Buda. Some optimal codes have structure. *IEEE Journal on Selected Areas in Communications*, 7(6):893–899, Aug. 1989.

A. El Gamal and Y. Kim. *Network Information Theory*. Cambridge University Press, 2011.

U. Erez and R. Zamir. Achieving 1/2log(1+snr) on the AWGN channel with lattice encoding and decoding. *IEEE Transactions on Information Theory*, 50(10):2293–2314, Oct. 2004.

I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Elsevier/Academic Press, Amsterdam, seventh edition, 2007.

W. Gu, R. Koetter, M. Effros, and T. Ho. On source coding with coded side information for a binary source with binary side information. 2007.

D. Gündüz, A. Yener, A. Goldsmith, and H. V. Poor. The multiway relay channel. *IEEE Transactions on Information Theory*, 59(1):51–63, Jan. 2013.

A. Kavčić, X. Ma, and M. Mitzenmacher. Binary intersymbol interference channel: Gallager codes, density evolution, and code performance bounds. *IEEE Transactions on Information Theory*, 49(7):1636–1652, 2003.

A. Khina, O. Ordentlich, U. Erez, Y. Kochman, and G. W. Wornell. Decode-and-forward for the Gaussian relay channel via standard AWGN coding and decoding. *IEEE Information Theory Workshop (ITW), Lausanne, Sweiss*, 2012.

G. Kramer, M. Gastpar, and P. Gupta. Cooperative strategies and capacity theorems for relay networks. *IEEE Transactions on Information Theory*, 51(9):3037–3063, Sept. 2005.

P. J. Lee. High-rate convolutional code construction with the minimum required SNR criterion. The telecommun. and data acquisition progr. rept., NASA JPL, Aug. 1985.

S. Lin and D. J. Costello. *Error control coding.* Prentice-Hall, 2004.

A. D. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6:440–442, 2002.

H. Loeliger. Averaging bounds for lattices and linear codes. *Transactions on Information Theory*, 43(6):1767–1773, Nov. 1997.

T. N. Morrissey. Analysis of decoders for convolutional codes by stochastic sequential machine methods. *IEEE Transactions on Information Theory*, 16(4):460–469, July 1970.

T. N. Morrissey. A Markovian analysis of Viterbi decoders for convolutional codes. *Proc. Nat. Electronics Conf.*, Oct. 1969.

W. Nam, S. Chung, and Y. H. Lee. Capacity bounds for two-way relay channels. *Int. Zurich Semin.*, 2008.

B. Nazer and M. Gastpar. Compute-and-forward: harnessing interference through structured codes. *IEEE Transactions on Information Theory*, 57(10):6463–6486, Oct. 2011.

M. Nokleby and B. Aazhang. Lattice coding over the relay channel. *IEEE International Confonference on Communication (ICC)*, 2011.

G. Poltyrev. On coding without restrictions for the AWGN channel. *IEEE Transactions on Information Theory*, 40(2):409–417, Mar. 1994.

B. Rankov and A. Wittneben. Achievable rate regions for the two-way relay channel. *IEEE International Symposium Information Theory (ISIT)*, 2006.

T. Richardson and R. Urbanke. Design of capacity-approaching irregular low-density parity check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001a.

T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001b.

C. A. Rogers. Lattice coverings of space. *Mathematica*, 6:33–39, 1959.

A. Savard and C. Weidmann. Décodeur amélioré pour le codage de source avec information adjacente compressée. *GRETSI*, 2013a.

A. Savard and C. Weidmann. Improved decoding for binary source coding with coded side information. *IEEE Information Theory Workshop (ITW), Sevilla, Spain*, 2013b.

A. Savard and C. Weidmann. Optimized codes for the binary coded side-information problem. *IEEE International Symposium on Turbo Codes (ISTC)*, 2014a.

A. Savard and C. Weidmann. On the multiway relay channel with direct links. *IEEE Information Theory Workshop (ITW), Hobart, Tasmania*, 2014b.

A. Savard and C. Weidmann. Canal à relais multidirectionnel avec liens directs. *GRETSI*, 2015a.

A. Savard and C. Weidmann. Lattice coding for the gaussian one- and two-way relay channels with correlated noises. *IEEE International Symposium Information Theory (ISIT)*, 2015b.

A. Savard and C. Weidmann. On the hamming-space voronoi regions of convolutional codes with applications. *submitted to IEEE Transactions on Communication*, 2015c.

A. Savard and C. Weidmann. On the multiway relay channel with intra-cluster links. *to be submitted to IEEE Transactions on Wireless Communications*, 2015d.

J. P. M. Schalkwijk, K. A. Post, and J. P. J. C. Aarts. On a method of calculing the event error probability of convolutional codes with maximum likelihood decoding. *IEEE Transactions on Information Theory*, 25(6):737–743, November 1979.

J. P. M. Schalkwijk, A. J. Vinck, and K. A. Post. Syndrome decoding of binary rate k/n convolutional codes. *IEEE Transactions on Information Theory*, 24(5):553–562, September 1978.

D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

S. Smirani, M. Kamoun, M. Sarkiss, A. Zaidi, and P. Duhamel. Achievable rate regions for two-way relay channel using nested lattice coding. *IEEE Trans. on Wireless Communications*, Jun. 2014.

Y. Song and N. Devroye. A lattice compress-and-forward scheme. *IEEE Information Theory Workshop (ITW)*, 2011.

Y. Song and N. Devroye. Lattice codes for the Gaussian relay channel: Decode-and-forward and compress-and-forward. *IEEE Transactions on Information Theory*, 59(8):4927 – 4948, Aug. 2013.

R. Storn and K. Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.

H. Tang and M. Lin. On $(n, n-1)$ convolutional codes with low trellis complexity. *IEEE Transactions on Communication*, 50(1):37–47, Jan. 2002.

N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *Proceedings on the 37th annual Allerton Conference on Communication, Control, and Computing*, 1999.

R. Urbanke and B. Rimoldi. Lattice codes can achieve capacity on the AWGN channel. *Transactions on Information Theory*, 44(1):273–278, Jan. 1998.

E. C. van der Meulen. Three-terminal communication channels. *Adv. Appl. Prob.*, 3:120–154, 1971.

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967.

R. Zamir and M. Feder. On lattice quantization noise. *IEEE Transactions on Information Theory*, 42(4):1152–1159, Jul. 1996.

L. Zhang, J. Jiang, A. J. Goldsmith, and S. Cui. Study of Gaussian relay channels with correlated noises. *IEEE Transactions on Information Theory*, 59(3):863–876, March 2011.

Coding for cooperative communications: Topics in distributed source coding and relay channels

Abstract: The current wireless data traffic growth cannot be handled by classical multi-hop network protocols as in interference-free wired networks, thus it has been recognized that network nodes need to cooperate in order to take advantage of source and/or channel signal correlations, which is needed to achieve fundamental capacity limits.
This thesis first considers a cooperative source coding problem, namely binary source coding with coded side information (CoSI): the helper node has access to a signal that is correlated with the source and may send a compressed version on a separate link to the destination, thus rate can be saved on the main source-destination link. Using a characterization of the Hamming-space Voronoi regions of the quantizer at the helper node, an improved practical scheme based on LDPC codes is proposed.
The second part of the thesis considers cooperative channel coding, where helper nodes are relays. The simplest example of such a communication is the relay channel, in which a relay node helps the source to send its message to the destination. Whereas in the source coding problem, the correlation between source and side information is given, in channel coding, the main question is to find the best relaying operation. First, a somewhat dual problem to source coding with CoSI is studied, by considering correlated noises at the relay and destination. Then, various extensions of the relay channel are characterized using upper bounds on capacity and achievable rates: the two-way relay channel with correlated noises at the relay and destinations, where two sources wish to exchange their data with the help of a relay, and the multiway relay channel with direct links, where users, grouped into fully connected clusters (users in a cluster can overhear each others' messages), wish to exchange their messages locally within a cluster with the help of one relay.

Keywords: Distributed source coding, Iterative decoding, Voronoi cells, Lattice coding, Relay channels

Codage pour les communications coopératives : Codage de source distribué et canaux à relais

Résumé : L'augmentation du trafic sur les réseaux sans fil ne permet plus de traiter les données en utilisant les protocoles standards des réseaux filaires, qui sont eux sans interférences. Ainsi, les noeuds des réseaux sans fil doivent coopérer en exploitant les corrélations inhérentes à la proximité des utilisateurs afin d'exploiter au mieux la capacité d'un tel réseau.
Dans cette thèse, nous considérons tout d'abord le problème de codage de source avec information adjacente compressée. Le noeud coopératif, ayant accès à un signal corrélé avec celui de la source, peut en envoyer une version compressée au destinataire sur un lien indépendant, permettant d'économiser du débit sur le lien principal. En utilisant une caractérisation des cellules de Voronoi du quantificateur utilisé, nous avons pu améliorer un algorithme de décodage itératif basé sur des codes LDPC.
La seconde partie de la thèse traite des problèmes de codage de canal, où les noeuds coopératifs sont des relais. L'exemple le plus simple d'une telle communication est le canal à relais, où un relais aide à la communication entre la source et la destination. Alors que dans le problème de codage de source, le canal de corrélation entre la source et le noeud coopératif est fixé, dans le codage de canal, la question est de savoir quelle opération effectuer au relais. Tout d'abord, nous considérons un problème quelque peu dual au problème de codage de source avec information adjacente compressée, en considérant des bruits corrélés au relais et la destination. Puis, nous étudions des bornes sur la capacité et des débits atteignables pour deux extensions du canal à relais, le canal à relais bidirectionnel avec des bruits corrélés au relais et aux destinations, où deux sources échangent leurs données avec l'aide d'un relais, et le canal multidirectionnel avec liens directs (qui modélisent la proximité des utilisateurs), où les utilisateurs sont regroupés dans des clusters et échangent leurs données localement au sein d'un même cluster avec l'aide d'un relais.

Mots clés : Codage de source distribué, Décodage itératif, Cellules de Voronoi, Codage par réseau de points, Canaux à relais