ulm university universität
uulm

Differential Linear Network Coding in Slowly-Varying Networks

Master Thesis
of
Carmen Maria Sippel

Institute of Communications Engineering
Ulm University
June 2018

M/2018/PU/01

ulm university universität
uulm

Institute of Communications Engineering

MASTER THESIS

Differential Linear Network Coding in Slowly-Varying Networks

Abstract

The topic of Network Coding (NC) is an emerging field of research in communication technology. It was introduced in [ACLY00] as a method of achieving the maximal throughput in a network. Due to the task of nodes in the network to randomly linear combine incoming messages over a finite field, the name Random Linear Network Coding (RLNC) was established. In [SKK08] a special form of rank-metric codes was proposed for error correction in RLNC networks. These rank-metric codes built up subspace codes, which could be used for error control in RLNC according to [KK08], by so-called lifting. This procedure has the drawback that the data rate is severely limited by its construction. Therefore a differential approach called Differential Linear Network Coding (DLNC) was introduced in [SCFH13], where higher rates could be achieved under the premise that the network does not considerably change between several transmissions. Further investigations into the tolerability of possible variations have been carried out in [PCF+15].

The aim of this work is to analyze both procedures in a probabilistic channel, mainly by comparing their failure probabilities. As a result, parameter regions are found, where DLNC outperforms lifting for certain network properties. The pure mathematical view is topped off with a simulative comparison on the error rates of both transmission schemes.

Submission date: June 1st, 2018

Candidate: Carmen Maria Sippel

Supervisor: Sven Puchinger

1st Examiner: Prof. Dr.-Ing. Martin Bossert

2nd Examiner: Prof. Dr. Irene Bouw

Catalog No.: M/2018/PU/01

I confirm, that the presented master thesis is an original work completed independently without inadmissible outside help other than the indicated sources. I also certify that this work has not been submitted to any other examination board in the same or similar form and all references, direct and indirect, are indicated as such and have been cited accordingly.

Ulm, June 1st, 2018

(Carmen Maria Sippel)

Contents

1. Introduction

Nowadays communications technology plays a major role in the lives of a vast amount of people, either directly, e.g. when using mobile communications for a phone call, or indirectly, e.g. with basic services that increasingly rely on global connectivity, such as supply chains. In particular digital communication networks are important for the exchange of information in these scenarios, as it can be seen in the example of the Internet. The leading principle for the distribution of information or the targeted consignment of messages is called routing. Other than in analog telephony, where the exchange of information happens continuously over once set-up circuit-switched links, in digital communication networks messages are usually encapsulated into packets and sent via determined paths.

With the results of Ahlswede et al. [ACLY00] the idea arose, that the throughput via a communication network can be extended to the maximum flow possible in the network. The basis for their work is the so-called *max-flow min-cut theorem* (cf. e.g. [CLRS09]). This is a result from graph theory, revealing that the capacity of the network, which is the min-cut, can be achieved by a so-called flow. This flow then represents the maximum throughput.

Ahlswede et al. converted the result for coding theory by specifying the rate region for a block code. While doing so the term Network Coding (NC) [ACLY00] was established. The conceptual innovation of their research was to not consider the transmission to be guided over single paths but randomly distributed, meaning that the information should spread in the network in a non-dedicated way. Apart from increased throughput, NC offers more benefits, such as security, robustness for link failures and packet loss (cf. [HL08], for security also [FS07]).

Another milestone in the topic of NC is [KK08], where a channel model, the so-called *operator channel* was introduced and the problem was tackled by subspace codes (also called codes in projective space). Proceeding from these subspace codes, Silva et al. proposed a technique called lifting [SKK08], which transfers rank-metric codes to a special class of subspace codes, namely constant-dimension codes. Furthermore, they introduced a matrix representation for the operator channel, which not only modeled the network but also errors imposed on edges in the network, and coined the term Random Linear Network Coding (RLNC), that is the application of codes

in order to cope with the errors in a network where the information is combined randomly at inner nodes. Meanwhile, applications, e.g. [KRH+06], and fields of use are suggested for RLNC with meshed wireless networks leading the way.

Another possibility for the application of rank-metric codes for error correction in a RLNC setting is Differential Linear Network Coding (DLNC) as introduced in [SCFH13]. The procedure rests upon the fact that communication networks usually do not change vastly and it provides better error correction capabilities especially for high data rates. The method is analogous to the Differential Phase-Shift Keying (DPSK) scheme, which is the differential version of Phase-Shift Keying (PSK). It has been shown that for DLNC, together with the demodulation step, a so-called Additive Matrix Channel (AMC) can be established [SCFH13]. A first intuitive analysis of this channel yields that, as with DPSK, the DLNC procedure has to cope with twice the amount of errors. [PCF+15] were the first to consider slow variations in the channel. They mainly proposed Partial Unit Memory (PUM) codes as means to overcome the changes in the network.

This work shall combine the results gained in [SCFH13] and [PCF+15] and attain more insight into the mechanics and behavior of the procedures of lifting and DLNC under the presumption of a probabilistic channel. The aforementioned publications miss an analytic examination of the DLNC scheme. This gap shall be filled by the presented work. The main contributions are statements about the relation of the failure probabilities of both methods in static, as well as in varying networks in combination with parameter ranges. To be precise, it is derived for which parameters the DLNC procedure outperforms lifted rank-metric codes in static networks and it is analyzed in how far the network is allowed to change without impairing DLNC so much that it cannot compete with the lifted rank-metric codes.

The thesis is structured as follows. Chapter 2 comprises the concepts that are necessary to understand the main ideas of lifting and DLNC, starting with codes in rank metric, continuing with the channel model for RLNC and concluding with the main aspects of DLNC.

In Chapter 3 we conduct some considerations on the rate and rank conditions, as well as on the use of so-called interleaved Gabidulin Codes in RLNC.

Then we start with the probabilistic analysis of static networks in Chapter 4, followed by Chapter 5, which is concerned with the analysis of varying networks.

The setup for the simulations can be found in Chapter 6, more precisely Sections 6.1 - 6.3. There it is described how the channel has been implemented and how decoding failures are recognized. The simulation results are located in Section 6.4.

In Chapter 7 a summary of the results as well as future work is presented.

2. Theory

Before starting to investigate the behavior of so-called lifting and DLNC, we introduce some known theory. Note that most of Sections 2.1 and 2.2 originates from [Wac13]. Hence we only mention it, if other sources were used or if a part is explicitly taken from [Wac13]. In Section 2.2 we introduce a concept called rank metric, which functions as fundament for rank-metric codes. In Section 2.3 the concept of RLNC and the system model is described.

2.1. Preliminaries

Throughout this work q is the power of a prime p, $\mathbb{F}_q$ is a finite field of order q, also called ground field. Since q is power of a prime, $\mathbb{F}_q$ is an extension field of degree $\ell := \log_p(q)$ and p is called the characteristic [LN83, Theorem 2.2]. However for our purposes an extension field of $\mathbb{F}_q$ of degree m, i.e. a m-dimensional vector space over $\mathbb{F}_q$ is necessary. This extension field is denoted by $\mathbb{F}_{q^m}$. If $a \in \mathbb{F}_{q^m}$ is a an element of the extension field, it can be represented as a vector over $\mathbb{F}_q$ using an arbitrary but fixed basis of $\mathbb{F}_{q^m}$ over the ground field. As for prime fields, the cardinality of the basis is m. Determining the order of the basis provides the opportunity to map $a \in \mathbb{F}_{q^m}$ onto the vector over $\mathbb{F}_q$. The formal construction of $\mathbb{F}_{q^m}$ works the same as the extension from $\mathbb{F}_p$ to $\mathbb{F}_q$, that is using an irreducible polynomial and one of its roots. So, combining elements of $\mathbb{F}_{q^m}$ to a row vector $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_{q^m}^n$ leads to the possibility of describing $\boldsymbol{a}$ as matrix $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ over the ground field by expressing every element of the row vector as (column) vector over $\mathbb{F}_q$. We denote the representation matrix as follows:

$$\boldsymbol{A} = \begin{pmatrix} A_{0,0} & A_{0,1} & \ldots & A_{0,n-1} \\ A_{1,0} & A_{1,1} & \ldots & A_{1,n-1} \\ \vdots & \vdots & & \vdots \\ A_{m-1,0} & A_{m-1,1} & \ldots & A_{m-1,n-1} \end{pmatrix}, \tag{2.1}$$

where the vector representation of one element of $\boldsymbol{a}$ is

$$a_j = \begin{pmatrix} A_{0,j} \\ A_{1,j} \\ \vdots \\ A_{m-1,j} \end{pmatrix} \in \mathbb{F}_q^{m\times 1} \qquad \forall j \in \{0,\ldots,n-1\}.$$

The mapping is given by [Wac13, Def. 2.1]

$$a_j = \sum_{i=0}^{m-1} A_{i,j}\beta_i \quad \forall j \in \{0,\ldots,n-1\},$$

where β_i are the fixed basis vectors of an arbitrary basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. As convention, small letters denote elements of $\mathbb{F}_{q^m}$, capital letters such over $\mathbb{F}_q$. Further we need the row space of a matrix.

Definition 2.1.1 (Row Space [Mey00])
The row space of a matrix $\boldsymbol{X}$ is defined by

$$\langle \boldsymbol{X} \rangle = \left\{ \boldsymbol{x} = \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i : \lambda_i \in \mathbb{F}_q, \boldsymbol{x}_i \text{ is the } i\text{-th row vector of } \boldsymbol{X} \right\}.$$

The following bounds on the rank of matrices are crucial for the most of the results. For reference reasons they are stated here. Let $\boldsymbol{A} \in \mathbb{F}_q^{N\times n}$ and $\boldsymbol{B} \in \mathbb{F}_q^{n\times M}$, then

$$\operatorname{rk}(\boldsymbol{A}\cdot\boldsymbol{B}) \leq \min\{\operatorname{rk}(\boldsymbol{A}), \operatorname{rk}(\boldsymbol{B})\} \tag{2.2}$$
$$\operatorname{rk}(\boldsymbol{A}+\boldsymbol{B}) \leq \operatorname{rk}(\boldsymbol{A}) + \operatorname{rk}(\boldsymbol{B}) \tag{2.3}$$
$$\operatorname{rk}(\boldsymbol{A}+\boldsymbol{B}) \geq |\operatorname{rk}(\boldsymbol{A}) - \operatorname{rk}(\boldsymbol{B})|. \tag{2.4}$$

More bounds can be found in [SKK08]. Also note that if $\boldsymbol{B}$ is of full rank, it is $\operatorname{rk}(\boldsymbol{A}\cdot\boldsymbol{B}) = \operatorname{rk}(\boldsymbol{A})$.

2.2. Rank-Metric Codes

Instead of using vectors in combination with the Hamming metric as codes, also matrices can be used. For these matrices, the above mentioned representation in $\mathbb{F}_q^{m\times n}$ of row vectors over $\mathbb{F}_{q^m}^n$ is used. A special code class emerged from this treatment. These are called rank-metric codes. Their definition is based on the concept of rank metric, which is introduced in the following paragraph.

2.2.1. Rank Metric

The notion of rank metric has been independently introduced by Delsarte [Del78], Gabidulin [Gab85] and Roth [Rot91]. The announced concept is based on the rank distance and rank weight, which are defined as follows.

Definition 2.2.1 (Rank Weight, Rank Distance [Gab85], [Del78], [Rot91]) *Let $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1}), \boldsymbol{b} = (b_0, b_1, \ldots, b_{n-1}) \in \mathbb{F}_{q^m}^n$ and $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{F}_q^{m \times n}$ their matrix representations over the ground field $\mathbb{F}_q$. Then the rank weight of $\boldsymbol{a}$ is defined to be the rank of its matrix representation $\boldsymbol{A}$:*

$$\mathrm{wt}_{\mathrm{rk}}(\boldsymbol{a}) \coloneqq \mathrm{rk}\,(\boldsymbol{A})$$

and the rank distance between the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is the rank of the difference of the matrix representations of both:

$$\mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{a}, \boldsymbol{b}) \coloneqq \mathrm{rk}\,(\boldsymbol{A} - \boldsymbol{B})\,.$$

Important to know is that the rank distance is a metric, since it fulfills the three conditions for metrics. Namely for $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{F}_{q^m}^n$ it is

1. non-negativity: $\mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{a}, \boldsymbol{b}) \geq 0 \ \wedge\ \mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{a}, \boldsymbol{b}) = 0 \iff \boldsymbol{a} = \boldsymbol{b}$,
2. symmetry: $\mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{a}, \boldsymbol{b}) = \mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{b}, \boldsymbol{a})$ and
3. triangle inequality: $\mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{a}, \boldsymbol{c}) \leq \mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{a}, \boldsymbol{b}) + \mathrm{dist}_{\mathrm{rk}}\,(\boldsymbol{b}, \boldsymbol{c})$.

If we now turn to rank-metric codes, some parameters are needed to describe them. These are the length n (number of codeword symbols per row), the cardinality $\mathcal{M}$, that is the number of codewords and the minimum rank distance, which is defined as follows:

Definition 2.2.2 (Minimum Rank Distance [Gab85], [Del78], [Rot91]) *Let $\mathcal{C}$ be a rank-metric code. Then its minimum rank distance is defined by*

$$d_{\mathrm{rk}} \coloneqq \min_{\substack{\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)} \in \mathcal{C} \\ \boldsymbol{c}^{(1)} \neq \boldsymbol{c}^{(2)}}} \left\{ \mathrm{dist}_{\mathrm{rk}}\left(\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}\right) \right\}.$$

Analogous to Hamming metric the minimum rank distance equals the minimum rank weight for linear codes.

2.2.2. Linearized Polynomials

Gabidulin codes are defined by evaluating so-called linearized polynomials. These are described as follows.

Definition 2.2.3 (Linearized Polynomials [Ore33])
A polynomial $p(x)$ is a linearized polynomial if

$$p(x) = \sum_{i=0}^{d_p} p_i x^{q^i}, \qquad p_i \in \mathbb{F}_{q^m} \quad \forall i \in 0, 1, \ldots, d_p,$$

where $p_{d_p} \neq 0$. Then $\deg_q(p(x)) = d_p$ is called the q-degree of $p(x)$.

Together with ordinary polynomial addition and the so-called symbolic product the linearized polynomials form a non-commutative univariate polynomial ring, which we be denote by $\mathbb{L}_{q^m}[x]$. Let us recall that a ring is a fundamental mathematical structure consisting of a set (here it is the set of linearized polynomials) and the two operations of addition and multiplication, where closure, associativity, distributivity and, in case of addition, commutativity hold. Furthermore for each operation there exist identities. In our case $x^{q^0} = x$ is the multiplicative identity and the all-zero polynomial is the additive identity.

Since the usual polynomial multiplication is not closed in the set of linearized polynomials, the symbolic product fills the role of the multiplication in the polynomial ring. The symbolic product is defined as the composition of two linearized polynomials and denoted by $a(x) \circ b(x) \coloneqq a(b(x))$. Let $c(x) = a(x) \circ b(x)$, then its coefficients c_j are calculated via

$$c_j = \sum_{i=0}^{j} a_i b_{j-i}^{q^i}, \qquad \forall j \in [0, d_a + d_b],$$

where d_a and d_b are the q-degrees of $a(x)$, respective $b(x)$. The adjective "linearized" stems from the fact, that the evaluation of a linearized polynomial is linear over $\mathbb{F}_q$.

Theorem 2.2.4 [LN83, p. 108]
Let $p(x) \in \mathbb{L}_{q^m}[x]$, $s_i \in \mathbb{F}_{q^m}$ and $A_i \in \mathbb{F}_q$ for $i \in \{1, 2\}$, then

$$p(A_1 s_1 + A_2 s_2) = A_1 p(s_1) + A_2 p(s_2). \tag{2.5}$$

Proof:
This is possible because $\forall A \in \mathbb{F}_q$ it holds that $A^{q^i} = A \ \forall i \in \mathbb{N}$. Furthermore in commutative rings with characteristic p it is $(a+b)^{p^i} = a^{p^i} + b^{p^i}$ for a, b in the ring,

i an integer [LN83, Theorem 1.46], which in turn causes $(a+b)^{q^i} = a^{q^i} + b^{q^i}$ in $\mathbb{F}_{q^m}$, since $q^i = p^{\ell^i}$ and ℓ^i is an integer. Thus allowing

$$\begin{aligned} p(A_1 s_1 + A_2 s_2) &= \sum p_i (A_1 s_1 + A_2 s_2)^{q^i} \\ &= A_1 \sum p_i s_1^{q^i} + A_2 \sum p_i s_2^{q^i} \\ &= A_1 p(s_1) + A_2 p(s_2). \end{aligned}$$

□

Therefore the evaluation of a linearized polynomial gets connected to the matrix representations used before. Having an element $s \in \mathbb{F}_{q^m}$ which can be represented by a (column) vector $\boldsymbol{S} = (S_0, S_1, \ldots, S_{m-1})^\top \in \mathbb{F}_q^{m\times 1}$ with the basis $\mathcal{B} = \{\beta_0, \ldots, \beta_{m-1}\}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ (the order of the basis has to be fixed), the evaluation of a linearized polynomial $p(x)$ over $\mathbb{F}_{q^m}$ is given by

$$p(s) = \sum_{i=0}^{m-1} S_i p(\beta_i). \tag{2.6}$$

With these definitions we can turn to Gabidulin codes, which are addressed in the following section.

2.2.3. Gabidulin Codes

Gabidulin codes are an important class of rank-metric codes. The definition utilizes the evaluation of a linearized polynomial with linearly independent elements over $\mathbb{F}_q$.

Definition 2.2.5 (Linear Gabidulin Code [Gab85], [Del78], [Rot91])
Let $g_0, g_1, \ldots, g_{n-1} \in \mathbb{F}_{q^m}$ be fixed elements and linearly independent over $\mathbb{F}_q$. Then a linear Gabidulin code $\mathcal{G}[n,k]$ over $\mathbb{F}_{q^m}$ of length $n \leq m$ and dimension $k \leq n$ is the set of all words, that fulfill

$$\mathcal{G}[n,k] \coloneqq \left\{ \big(f(g_0),\, f(g_1),\, \ldots,\, f(g_{n-1})\big) : f(x) \in \mathbb{L}_{q^m}[x], \deg_q f(x) < k \right\},$$

i.e. the set of evaluation points of the above defined fixed elements $g_0, g_1, \ldots, g_{n-1}$.

As evaluation points one could, for example, choose a *normal basis*, since the computations can be done efficiently with such a basis [Wac13, Section 3.1.1]. A normal basis is characterized by the fact, that $\beta_i = \beta^{q^i}$ for all β_i in the basis, i.e. the basis is generated by one so-called *normal element* $\beta \in \mathbb{F}_{q^m}$. Using the definition of the q-transform from [Wac13, Definition 2.12] the Gabidulin codewords can be seen as

the inverse q-transform of the evaluation polynomial $f(x)$. This view is analogous to the definition of Reed Solomon (RS) codes by the Discrete Fourier Transform (as it is done e.g. in [Bos13, Definition 3.4]). These Gabidulin codes are then q-cyclic. The structure of Gabidulin codes rewards them, being the RS code equivalent in rank metric. Considered as codes in Hamming metric, Gabidulin codes are (as RS codes) maximum distance separable, which means they are optimally spread in the space they are defined in. The rank metric equivalent for Maximum Distance Separable (MDS) codes are Maximum Rank Distance (MRD) codes. In rank metric, Gabidulin codes are also MRD codes, because they fulfill the rank metric equivalent of the Singleton bound with equality, i.e. the minimum rank distance of a linear Gabidulin code achieves $d_{\mathrm{rk}} = n - k + 1$.

The generator matrix of a Gabidulin code can be calculated as follows

$$\boldsymbol{G}_{\mathcal{G}} = \begin{pmatrix} g_0^{q^0} & g_1^{q^0} & \cdots & g_{n-1}^{q^0} \\ g_0^{q^1} & g_1^{q^1} & \cdots & g_{n-1}^{q^1} \\ \vdots & \vdots & & \vdots \\ g_0^{q^{k-1}} & g_1^{q^{k-1}} & \cdots & g_{n-1}^{q^{k-1}} \end{pmatrix},$$

cf. [Gab85, Theorem 7]. Multiplication of coefficients u_i to the row of the generator matrix is the same as evaluating the linearized polynomial, which defines the Gabidulin code, i.e. if $\boldsymbol{u} \in \mathbb{F}_{q^m}^k$, then $\boldsymbol{c} = \boldsymbol{u} \cdot \boldsymbol{G}_{\mathcal{G}} \in \mathcal{G}[n, k]$.

Error-Erasure Decoding of Gabidulin Codes

Error-erasure decoding utilizes insight into the error, obtained via side information. This can be explained as follows. An error always consists of an error value and its location in the codeword. Since matrices are used as codewords, these terms can be matched to column space and row space of the error matrix. In case the column space is known, one usually speaks of an erasure or column erasure (cf. [Wac13]). For the contrary case, when the row space is known, so-called deviations were introduced in [SKK08]. Here [Wac13] used the term row erasure. The last case is that neither is known, which ends up in a full error. Table 2.1 shows an overview on all notations on errors, erasures and their numbers. These erasures and deviations provide partial inside into the errors. Figure 2.1 extracts erasures, deviations and full errors from the error matrix. $\boldsymbol{L} \cdot \hat{\boldsymbol{E}}$ represent the erasures, where the column space (gray) is known. The second addend relates to the deviations, i.e. the row space $\hat{\boldsymbol{L}}$ is known.

Silva et al. showed in [SKK08, Theorem 11] under which conditions error-erasure decoding is possible.

Silva [SKK08]		Wachter-Zeh [Wac13]		here
notation	#	notation	#	#
full error	ϵ	(full) error	t	ϵ
erasure	μ	column erasure	γ	μ_{c}
deviation	δ	row erasure	ϱ	μ_{r}

Table 2.1.: Notation of errors and numbers of errors in [SKK08], [Wac13] and this work.

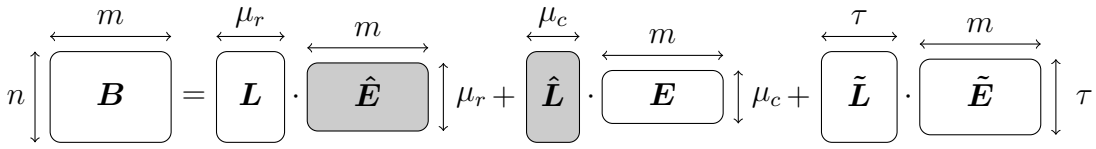


Figure 2.1.: Decomposition of the error matrix $\boldsymbol{B}$ into full-rank matrices, the grey matrices are known.

Theorem 2.2.6 [SKK08, Theorem 11]
Let ϵ the number of full errors, μ_{c} the number of erasures and μ_{r} the number of deviations in a received word, then error-erasure decoding of a Gabidulin code with minimum distance d_{rk} is possible if and only if

$$2\epsilon + \mu_{\mathrm{c}} + \mu_{\mathrm{r}} \leq d_{\mathrm{rk}} - 1. \tag{2.7}$$

They also gave algorithms for the decoding procedure. Noteworthy in equation (2.7) is that "erasures and deviations cost half of an error in the rank metric" [SKK08].

2.3. Error Correction in Random Linear Network Coding

In 2000 Ahlswede et al. proposed in [ACLY00] that for every communication network it is possible to achieve the maximum throughput with a strategy referred to as Network Coding (NC). Later a concept called RLNC, built upon the results of Ahlswede, was introduced by Kschischang et al. [KK08] and then combined with rank-metric codes in [SKK08].

In this work we regard the unicast scenario, so there is one sender and one receiver somewhere in the network and information shall be passed from the first to the latter. The nodes on the way between sender and receiver are called internal nodes,

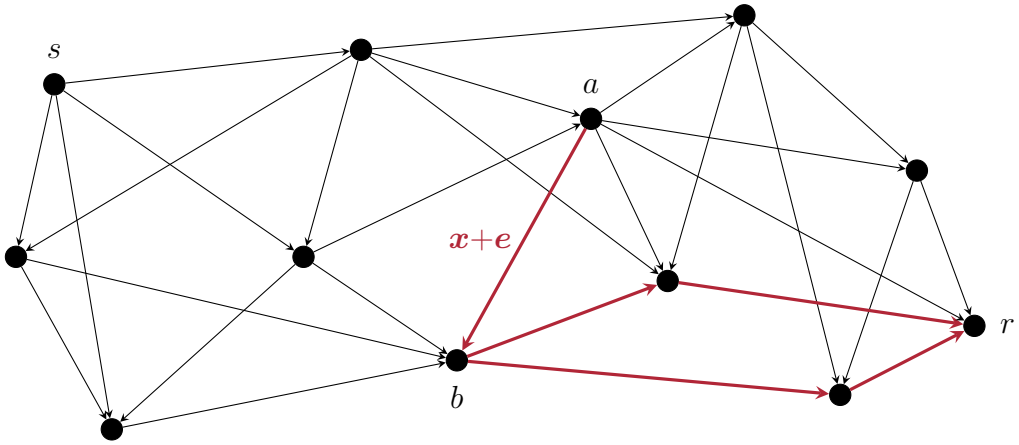


Figure 2.2.: Example of a considered network with sending node s and receive node r and the propagation of one error $\boldsymbol{e}$. $\boldsymbol{x}$ is a message vector sent between two inner nodes a and b.

they have the task to combine incoming packets using random linear factors (in case of RLNC) and spread them further. They might not be aware of the actual code, used by sender and receiver. An example of such a network is shown in Figure 2.2.

The information on a link is sent as (row) vectors $\boldsymbol{x} \in \mathbb{F}_q^M$, which are called packets. At the sender n packets are sent on different links at a time. So the information can virtually be packed together into a matrix $\boldsymbol{X} \in \mathbb{F}_q^{n \times M}$. The event of starting to send an information matrix up to the reception of it at the receiver side, is called generation. If only one generation is regarded, it is called a one-shot scenario, if more happen, it is usually referred to as a multi-shot scenario. Hence referring to a certain shot, is the same as talking about one generation. Another given condition is that the topology of the network is not known. Thus the transmission scheme is independent of it. Later there will be considerations on a changing topology between different generations in a multi-shot scenario. All assumptions and properties for the network can be found in Table 2.2.

Whenever information is sent over a link between two nodes, it can happen, that an error is superimposed. This error propagates through the network in the same way as it is done with the information packets, which means in case of RLNC, there will be linear combinations of it, i.e. one error spreads all over the network. Because these errors might reach the receiver by several paths, the network is quite sensitive to these transmission errors. Therefore, a suited method of error correction is necessary.

Kötter et al. [KK08] showed how subspace codes can be used advantageously in a RLNC scenario and in [SKK08] Silva et al. explained, how to construct subspace

definitions	description
internal nodes	nodes which are not addressed, but pass on the message as linear combination of the incoming messages
generation, shot	entity of a transmission between sending and receiving node
properties	**description**
point-to-point transmission/ unicast scenario	sending node addresses one receiving node
non-coherence	unawareness of the topology of the network
multi-shot network	consideration of more than one generation
time-varying	the network may change between generations (from Chapter 5 on)
labeling	packets of different generations can be distinguished and are processed separately (this is actually not a property but a treatment)

Table 2.2.: Definitions and properties of RLNC networks considered in this work.

codes in rank metric for the error correction in the same scenario. In the following the mathematical understanding of the network as channel is presented.

2.3.1. Channel Model

In [KK08] the operator channel was introduced, which compressed all the properties into a simple channel model. But since Kötter and Kschischang were working with subspace codes instead of rank-metric codes, the channel is a slightly modified for our purposes. Regard the transmit matrix $\boldsymbol{X} \in \mathbb{F}_q^{n \times M}$. The idea of RLNC is to build random linear combinations of incoming packets at each internal node and send them to all adjacent nodes [SKK08]. Note that for simplicity we regard only directed acyclic networks. The random factors of the linear combinations are chosen once and stay the same for one node, as long as it is in the network.

End-to-end consideration of these random linear combinations yields the multiplication with a random matrix $\boldsymbol{A}$, which is called the network channel matrix and stays the same as long as the network topology is not changed. It is an $N \times n$-matrix if N is the number of collected packets at the receiving side. Since the matrix multiplication is a linear transformation, it preserves the row space of the transmitted matrix, i.e. the span of its rows. This is an important feature for the use of codes

in network coding. However we did not consider errors in the channel up to now. That is why we introduce an $N \times M$-matrix $\boldsymbol{B}$, the additive error matrix, which emerges from the (possibly linearly combined) errors, that happen on links between nodes. The Hamming matrix of this error matrix might be high, but usually the rank is not. That is why it is useful to apply rank-metric codes in these scenarios. The incoming packets at the receiving node are denoted $\boldsymbol{Y}$ and calculate to

$$\boldsymbol{Y} = \boldsymbol{A} \cdot \boldsymbol{X} + \boldsymbol{B} \in \mathbb{F}_q^{N \times M}. \tag{2.8}$$

Due to the multiplication and addition of matrices this channel model is called Multiplicative Additive Matrix Channel (MAMC). If $\operatorname{rk}(\boldsymbol{Y}) < N$, the linear dependent packets are removed, since they do not give further information for the decoding problem. Therefore the receive matrix $\boldsymbol{Y}$ might not have the same size as it is stated above, but we can be sure it has full rank. For simplicity we assume $N = n$. All in all, what is obtained is a channel defining an RLNC network by the MAMC, Silva et al. called it Random Linear Network Channel (RLNCC).

Definition 2.3.1 (RLNC Network, cf. [SKK08, eq. (17)])
Let $\boldsymbol{A} \in \mathbb{F}_q^{N \times n}$, $\boldsymbol{B} \in \mathbb{F}_q^{N \times M}$ and $\boldsymbol{X} \in \mathbb{F}_q^{n \times M}$, then

$$\boldsymbol{Y} = \boldsymbol{A} \cdot \boldsymbol{X} + \boldsymbol{B}$$

defines the RLNC network. $\boldsymbol{A}$ is called channel matrix, $\boldsymbol{B}$ is the error matrix, $\boldsymbol{X}$ is the transmit matrix and $\boldsymbol{Y} \in \mathbb{F}_q^{N \times M}$ is the receive matrix.

The following lemma provides more insight into properties of the channel model. Since the proof in [SKK08] is presented from a different point of view, we propound another proof here.

Lemma 2.3.2 [SKK08, Lemma 14]
Let $\boldsymbol{A} \cdot \boldsymbol{X} + \boldsymbol{B}$ specify a RLNC channel with $\boldsymbol{X} \in \mathbb{F}_q^{n \times M}$ as the transmit matrix, $\boldsymbol{B} \in \mathbb{F}_q^{N \times M}$, the error matrix in one generation and $\boldsymbol{A} \in \mathbb{F}_q^{N \times n}$ the channel matrix. Then

$$\operatorname{rk} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{B} \end{bmatrix} = \operatorname{rk}(\boldsymbol{X}) + \operatorname{rk}(\boldsymbol{B}). \tag{2.9}$$

Proof:
The statement is equivalent to $\dim(\langle \boldsymbol{X} \rangle \cap \langle \boldsymbol{B} \rangle) = 0$. Assume one of the rows of $\boldsymbol{B}$ lies in the row space of $\boldsymbol{X}$, i.e. $\langle \boldsymbol{X} \rangle \cap \langle \boldsymbol{B} \rangle \neq \emptyset$, one could find a $\boldsymbol{B'}$ of lower rank, s.t. $\dim(\langle \boldsymbol{X} \rangle \cap \langle \boldsymbol{B'} \rangle) = 0$ and $\langle \boldsymbol{B'} \rangle \subset \langle \boldsymbol{B} \rangle$. Choose rank of $\boldsymbol{B'}$ to be maximal. More precisely construct $\boldsymbol{B'}$ from $\boldsymbol{B}$, by changing the row vectors $\boldsymbol{b}_i \in \langle \boldsymbol{X} \rangle$ from $\boldsymbol{B}$ to

$\boldsymbol{b}'_i = \mathbf{0}$ and keep all other row vectors. Then

$$\begin{aligned}\langle \boldsymbol{B} - \boldsymbol{B}' \rangle &= \left\{ \boldsymbol{b} = \sum_{i=1}^{n} \lambda_i (\boldsymbol{b}_i - \boldsymbol{b}'_i) : \lambda_i \in \mathbb{F}_q \right\} \\ &= \left\{ \boldsymbol{b} = \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i : \lambda_i \in \mathbb{F}_q, \boldsymbol{b}_i \neq \boldsymbol{b}'_i \right\} \\ &= \left\{ \boldsymbol{b} = \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i : \lambda_i \in \mathbb{F}_q, \boldsymbol{b}_i \in \langle \boldsymbol{X} \rangle \right\}.\end{aligned}$$

Concluding $\langle \boldsymbol{B} - \boldsymbol{B}' \rangle \subseteq \langle \boldsymbol{X} \rangle$, which implies that the subtraction of $\boldsymbol{B}$ and $\boldsymbol{B}'$ can be expressed as a transformed $\boldsymbol{X}$, say

$$\boldsymbol{B} - \boldsymbol{B}' = \boldsymbol{A}' \boldsymbol{X}. \tag{2.10}$$

Then the channel model equals another channel

$$\boldsymbol{A}\boldsymbol{X} + \boldsymbol{B} \overset{(2.10)}{=} \boldsymbol{A}\boldsymbol{X} + \boldsymbol{A}'\boldsymbol{X} + \boldsymbol{B}' = (\boldsymbol{A} + \boldsymbol{A}')\boldsymbol{X} + \boldsymbol{B}',$$

represented by a channel matrix $\boldsymbol{A} + \boldsymbol{A}'$ and an error matrix $\boldsymbol{B}'$ of lower rank than before. This should then be the channel to be considered. □

Accordingly the error matrix $\boldsymbol{B}$ should always be of smallest rank for given $\boldsymbol{A}, \boldsymbol{X}$.

2.3.2. Probabilistic Error Model

The errors on the edges can be seen as independent Bernoulli trials with different probabilities p_i.[1] Let $\bar{p} = \frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} p_i$, where $|\mathcal{N}|$ is the number of nodes in the network. Then the sum of these random variables can be approximated by the Binomial distribution with parameters $|\mathcal{N}|$ and $\bar{p}$. Choi [CX02] showed that the Binomial distribution is better than the Poisson distribution $\mathrm{Pois}(\bar{p})$ for finite $|\mathcal{N}|$. One can therefore assume that $\mathrm{rk}(\boldsymbol{B}_i) \sim \mathrm{Bin}(|\mathcal{N}|, \bar{p})$. Nevertheless the rank of the error matrix $\boldsymbol{B}_i$ must be smaller than the number of received packets (i.e. its first dimension). Therefore Seidl [SCFH13] proposes, that

$$\Pr\{\mathrm{rk}(\boldsymbol{B}_i) = k\} \approx 0, \forall k > n \iff \bar{p} \cdot |\mathcal{N}| \ll n$$

and can therefore be neglected. RLNC networks have a number of nodes, fulfilling $|\mathcal{N}| \gg n$, therefore for probabilistic analysis we assume $\mathrm{rk}(\boldsymbol{B}_i) \sim \mathrm{Bin}(n, p_\mathrm{B})$ with a respective $p_\mathrm{B} = \frac{|\mathcal{N}|}{n}\bar{p}$ in order to have the same expected value in both distributions. For completeness one can define $\Pr\{\mathrm{rk}(\boldsymbol{B}_i) = n\} = \sum_{i=n}^{|\mathcal{N}|} \binom{|\mathcal{N}|}{i} \bar{p}^i (1-\bar{p})^{|\mathcal{N}|-i}$, i.e. the Probability Mass Function (PMF) being clipped.

[1] p_i being the probability that an error happens on a particular edge.

2.3.3. Lifted Rank-Metric Codes

In order to cope with the sort of channel introduced in Definition 2.3.1 Silva et al. suggested lifted rank-metric codes (in fact lifted Gabidulin codes). The lifting construction utilizes an identity matrix to display parts of the network channel matrix $\boldsymbol{A}$, which otherwise would not be known to the receiver. For rank-metric codes it is defined as follows.

Definition 2.3.3 (Lifting Construction [SKK08, Def. 3])
Let $\boldsymbol{I}_{n\times n}$ be an identity matrix of size $n \times n$, $M = n+m$ and $\boldsymbol{S} \in \mathbb{F}_q^{n\times m}$ an arbitrary matrix. Then the function $\mathcal{L} : \mathbb{F}_q^{n\times m} \to \mathbb{F}_q^{n\times M}$, mapping

$$\boldsymbol{S} \mapsto \mathcal{L}(\boldsymbol{S}) = [\, \boldsymbol{I}_{n\times n} \mid \boldsymbol{S} \,],$$

is called lifting.

The matrix $\boldsymbol{S}$ might be coded or uncoded. Since coding is necessary for error control in random linear networks, the information is usually be encoded. Then it is denoted by a matrix $\boldsymbol{C}$. Obviously the action of lifting describes concatenating the actual codeword to a $n \times n$ identity matrix. In case Gabidulin codes are used for lifting, the transpose of a usual Gabidulin codeword-matrix is taken, in order to get a matrix from $\mathbb{F}_q^{n\times m}$. This has the benefit that the appended identity matrix has a smaller size due to $n \leq m$. Such a codeword is then called lifted Gabidulin codeword. In [Wac13, Lemma 2.18] it has been shown that lifted Gabidulin codes are MRD codes. The code is due to its construction a so-called constant-dimension code, which itself is a special class of subspace codes (also called codes in projective space). The lifted Gabidulin code has minimum subspace distance

$$d_{\text{Lifted}} = 2d_{\text{rk}} = 2(n - k + 1),$$

cf. [SKK08, Proposition 4]. The subspace distance is an important measure for subspace codes and defined as follows.

Definition 2.3.4 (Subspace Distance [SKK08, Def. 2])
Let $\boldsymbol{X} \in \mathbb{F}_q^{n\times M}$, $\boldsymbol{Y} \in \mathbb{F}_q^{N\times M}$. Then the subspace distance of their row spaces is defined to be

$$\begin{aligned} d_{\text{s}}(\langle \boldsymbol{X} \rangle, \langle \boldsymbol{Y} \rangle) &\coloneqq \dim(\langle \boldsymbol{X} \rangle + \langle \boldsymbol{Y} \rangle) - \dim(\langle \boldsymbol{X} \rangle \cap \langle \boldsymbol{Y} \rangle) \\ &= 2\dim(\langle \boldsymbol{X} \rangle + \langle \boldsymbol{Y} \rangle) - \dim(\langle \boldsymbol{X} \rangle) - \dim(\langle \boldsymbol{Y} \rangle). \end{aligned} \tag{2.11}$$

The subspace distance is also useful to rewrite the main result for the decoding capability of the lifted rank-metric codes as stated in [SKK08].

Decoding guarantee of the Lifting Construction

The following derivation stems in parts from a draft by Sven Puchinger. We regard the above introduced channel and a lifted rank-metric codeword $\boldsymbol{X}$ and the receive word $\boldsymbol{Y}$ of a RLNC channel as defined in 2.3.1. For the theoretic analysis of the error correction capabilities, we regard the subspace distance of the row spaces of transmit matrix $\boldsymbol{X}$ and receive matrix $\boldsymbol{Y}$ in combination with the result of Silva et al. as stated in (2.7).

Theorem 2.3.5 [SKK08, Theorem 9&10]
Let $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B} \in \mathbb{F}_q^{n\times M}$ be a receive matrix in a RLNC network with $\boldsymbol{A} \in \mathbb{F}_q^{n\times n}$ as channel matrix, $\boldsymbol{B} \in \mathbb{F}_q^{n\times M}$ as error matrix and $\boldsymbol{X} \in \mathbb{F}_q^{n\times M}$ as lifted rank-metric codeword, where the respective rank-metric code has minimum distance d_{rk}. Then decoding is possible if and only if

$$2\,\mathrm{rk}\,(\boldsymbol{B}) + n - \mathrm{rk}\,(\boldsymbol{Y}) \leq d_{\mathrm{rk}} - 1.$$

Proof:
Examination of Theorem 9 and the proof of Theorem 11 (the theorem is recited here as Theorem 2.2.6) in [SKK08] reveals, that error correction is possible if and only if

$$d_{\mathrm{s}}(\langle \boldsymbol{X}\rangle, \langle \boldsymbol{Y}\rangle) \overset{\text{Theorem 9}}{=\!=\!=} 2(\epsilon + \mu_{\mathrm{r}} + \mu_{\mathrm{c}}) - \mu_r - \mu_{\mathrm{c}} \overset{\text{Theorem 11}}{\leq} d_{\mathrm{rk}} - 1$$

with d_{rk} as defined above, ϵ the number of full errors, μ_{c} the number of (column) erasures and μ_{r} the number of row erasures (deviations in [SKK08]). It is known, that the rank of a matrix equals the dimension of its row space, i.e. $\mathrm{rk}\,(\boldsymbol{X}) = \dim(\langle \boldsymbol{X}\rangle)$. Furthermore it is

$$\mathrm{rk}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{Y}\end{bmatrix} = \dim(\langle \boldsymbol{X}\rangle + \langle \boldsymbol{Y}\rangle), \tag{2.12}$$

as it can be seen in equation (6) of [SKK08]. By inspecting the subspace distance we obtain:

$$\begin{aligned} d_{\mathrm{s}}(\langle \boldsymbol{X}\rangle, \langle \boldsymbol{Y}\rangle) &\overset{(2.11)}{=} 2\dim(\langle \boldsymbol{X}\rangle + \langle \boldsymbol{Y}\rangle) - \dim(\langle \boldsymbol{X}\rangle) - \dim(\langle \boldsymbol{Y}\rangle) \\ &\overset{(2.12)}{=} 2\,\mathrm{rk}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{Y}\end{bmatrix} - \mathrm{rk}\,(\boldsymbol{X}) - \mathrm{rk}\,(\boldsymbol{Y}) \qquad (2.13) \\ &\overset{(2.8)}{=} 2\,\mathrm{rk}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}\end{bmatrix} - \mathrm{rk}\,(\boldsymbol{X}) - \mathrm{rk}\,(\boldsymbol{Y}) \\ &\overset{\substack{\text{lower part}\\ -\boldsymbol{A}\boldsymbol{X}}}{=\!=\!=} 2\,\mathrm{rk}\begin{bmatrix}\boldsymbol{X}\\ \boldsymbol{B}\end{bmatrix} - \mathrm{rk}\,(\boldsymbol{X}) - \mathrm{rk}\,(\boldsymbol{Y}) \qquad (2.14) \end{aligned}$$

With assumption (2.9) from Lemma 2.3.2 we have

$$\begin{aligned} d_{\mathrm{s}}(\langle \boldsymbol{X} \rangle, \langle \boldsymbol{Y} \rangle) &\overset{(2.14)}{=} 2 \operatorname{rk} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{B} \end{bmatrix} - \operatorname{rk}(\boldsymbol{X}) - \operatorname{rk}(\boldsymbol{Y}) \\ &\overset{(2.9)}{=} 2 \operatorname{rk}(\boldsymbol{B}) + \operatorname{rk}(\boldsymbol{X}) - \operatorname{rk}(\boldsymbol{Y}) \\ &\overset{\operatorname{rk}(\boldsymbol{X})=n}{=\!=\!=} 2 \operatorname{rk}(\boldsymbol{B}) + n - \operatorname{rk}(\boldsymbol{Y}) . \end{aligned}$$

□

The result of Theorem 2.3.5 can be lower bounded by

$$d_{\mathrm{s}}(\langle \boldsymbol{X} \rangle, \langle \boldsymbol{Y} \rangle) \geq 2 \operatorname{rk}(\boldsymbol{B}) ,$$

since $\operatorname{rk}(\boldsymbol{X}) = n$ and $\operatorname{rk}(\boldsymbol{AX} + \boldsymbol{B}) \leq n$. Concluding that in the best case one can decode a lifted rank-metric code in a RLNC scenario as long as $\operatorname{rk}(\boldsymbol{B}) \leq \frac{d_{\mathrm{rk}}-1}{2}$.

2.4. Differential Linear Network Coding

If a lifted Gabidulin code is applied in a RLNC scenario, the channel matrix $\boldsymbol{A}$ is sounded in each generation, which implies that the network could change completely between two shots without impairing the success of the scheme. A vast change in the network is usually not the case. Therefore Differential Linear Network Coding (DLNC) was introduced in [SCFH13]. The method can be explained via DPSK, which is the differential version of Phase-Shift Keying (PSK). In PSK each modulated information point has a certain phase and exactly this phase information is transmitted. On the other hand, in DPSK the first point is fixed and only the phase differences between information points is sent. This is in particular useful if the phase change induced by the channel is not too high.

Analogous thereto DLNC has an initialization matrix and further information matrices are multiplied onto each other step after step, yielding a differential modulation scheme. For the network coding approach this means to start with an initial matrix $\boldsymbol{X}_0 = [\boldsymbol{I}_{n\times n} \,|\, \boldsymbol{0}_{n\times m}] \in \mathbb{F}_q^{n\times M}$, here $\boldsymbol{0}_{n\times m}$ is the $n \times m$-all-zero matrix, and multiply it with the information $\boldsymbol{S}_1 \in \mathbb{F}_q^{n\times M}$, $M = m + n \geq 2n$ (which could be a transposed Gabidulin codeword). In each shot the information matrix $\boldsymbol{S}_i \in \mathbb{F}_q^{n\times M}$ is multiplied to the previous transmit matrix, such that the current transmit matrix relates to the preceding one in the following way:

$$\boldsymbol{X}_i = (\boldsymbol{X}_{i-1})_{[n]} \cdot \boldsymbol{S}_i. \tag{2.15}$$

The notation $(\cdot)_{[n]}$ shall say that only the first n columns are used. A condition on $\boldsymbol{S}_i$ is that it has to have full rank in the front part, which leads to a loss $\sim 1/q$

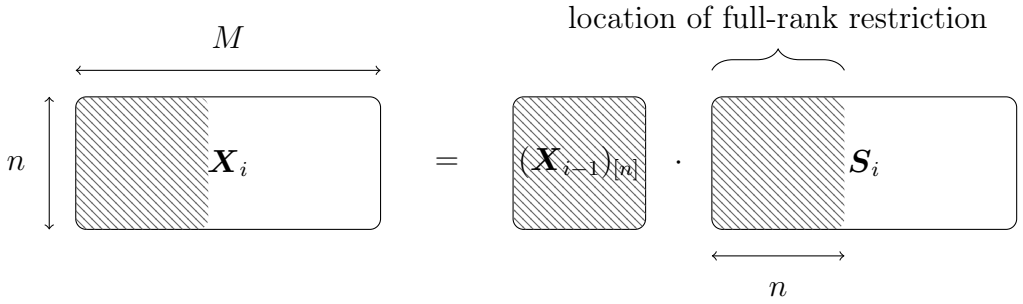


Figure 2.3.: Rank condition for the, possibly encoded, information matrix $\boldsymbol{S}_i$. The shaded parts shall have full rank.

(cf. [SCFH13]). The condition is illustrated in Figure 2.3. Since in matrix multiplications the second dimension of the first matrix has to be equal to the first dimension of the second matrix, we have to adjust the encoding for non-square matrices, i.e. take first n columns of the preceding sending matrix $\boldsymbol{X}_{i-1}$.

Due to the fact that of the $M = n + m$ columns n must be chosen to be invertible, Seidl introduced the loss L_{DLNC} (cf. [SCFH13, equation (23)])

$$L_{\text{DLNC}} = \frac{n}{q(n+m)}. \tag{2.16}$$

Note that $\text{GL}_n(\mathbb{F}_q)$ the general linear group, i.e. the set of all invertible matrices, is like the name says a group over matrix multiplications. Therefore it is closed, which leads to the fact, that all $\boldsymbol{X}_i$ are invertible, if $\boldsymbol{S}_i$ is always chosen to be invertible in the front part. The demodulation rests upon the matrix multiplication of the received matrix $\boldsymbol{Y}_i$ and its preceding (weak) pseudo inverse $\boldsymbol{Y}_{i-1}^+$. Such a weak pseudo inverse $\boldsymbol{Y}_i^+$ must fulfill

$$\boldsymbol{Y}_i^+ \cdot \boldsymbol{Y}_i = \boldsymbol{I}_n + \boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top. \tag{2.17}$$

Hereby $\mathcal{U} \subseteq \{0, 1, \ldots, n\}$ and $\boldsymbol{L}$ is a matrix that meets $\boldsymbol{I}_{\mathcal{U}}^\top \boldsymbol{L} = -\boldsymbol{I}_{|\mathcal{U}|}$ ($\boldsymbol{I}_{\mathcal{U}}$ equals $\boldsymbol{I}_{n\times n}$ without the rows not in $\mathcal{U}$). Silva et al. proved in [SKK08], that it is always possible to find such a matrix. The pseudo inverse is necessary because $\boldsymbol{A}$ might not be invertible. For non-square receive matrices, only the first n columns of $\boldsymbol{Y}_{i-1}$ have to be considered for the calculation of the pseudo inverse. So the demodulations calculates to

$$\hat{\boldsymbol{S}}_i = (\boldsymbol{Y}_{i-1})_{[n]}^+ \cdot \boldsymbol{Y}_i. \tag{2.18}$$

It has been shown in [SCFH13], that the demodulation result is obtained as the superposition of the transmitted information and an error, i.e. $\hat{\boldsymbol{S}}_i = \boldsymbol{S}_i + \boldsymbol{E}_i$, where the effective error matrix $\boldsymbol{E}_i$ depends on the current as well as the previous error matrix, the beforehand defined pseudo inverse $\boldsymbol{Y}_{i-1}^+$ and the transmission matrix

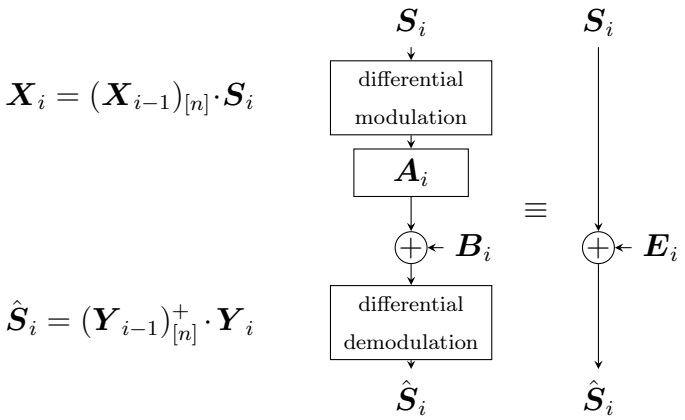


Figure 2.4.: Transformation of the differential modulation scheme to AMC according to [SCFH13].

$\boldsymbol{S}_i$. Thus one is concerned with an AMC, as illustrated in Figure 2.4, which can be handled using rank-metric codes like introduced in Chapter 2.2. However a first intuition shows, that one has to deal with an error rank at least two times higher than with the lifting construction, since the current as well as the previous error matrix influence the effective error matrix.

2.4.1. Varying Networks

The concept was enhanced in [PCF+15], allowing slow variations in the network, i.e. nodes joining or leaving between generations. This affects the network channel matrix $\boldsymbol{A}$. The changes can be expressed by the so-called channel deviation, which is defined as follows.

Definition 2.4.1 (Channel Deviation [PCF+15])
Let $\boldsymbol{A}_i$ be the channel matrix in the i-th generation, then we define the channel deviation

$$\Delta \boldsymbol{A}_i \coloneqq \boldsymbol{A}_i - \boldsymbol{A}_{i-1}$$

to be the change between current and preceding channel matrix.

It was proved in [PCF+15], that the rank of the channel deviation can be upper bounded by the minimum of incoming or outgoing edges of leaving nodes.

Theorem 2.4.2 [PCF+15, Theorem 1]
Let $\Delta \boldsymbol{A}_i$ be the channel deviation of an RLNC network in generation i. Further let ν be the only leaving or joining node between generation $i-1$ and i, as well as w_{in} be the number of its incoming edges and w_{out} the number of its outgoing edges. Then

$$\operatorname{rk}(\Delta \boldsymbol{A}_i) \leq \min\{n, w_{\text{in}}, w_{\text{out}}\}.$$

This can be comprehended by interpreting each new/missing linear combination sent to or from the appearing/disappearing node to other network codes as an error. The rank of the channel deviation is then always be smaller or equal than the number of incoming edges of this node (determining the number of incoming linear combinations) and also smaller than or equal to the outgoing edges (determining the outgoing linear combinations) of this leaving or joining node.

Furthermore detailed investigations into the PMF of $\operatorname{rk}(\Delta \boldsymbol{A}_i)$ were made. It is stated, that the PMF of $\operatorname{rk}(\Delta \boldsymbol{A}_i)$ can be upper bounded as follows. We define the respective random variable (RV) as $\overline{\operatorname{rk}(\Delta \boldsymbol{A}_i)}$.

$$\operatorname{rk}(\Delta \boldsymbol{A}_i) \leq \overline{\operatorname{rk}(\Delta \boldsymbol{A}_i)} \coloneqq \sum_{j=1}^{\ell} w(\nu_j),$$

where $\nu_j,\ j \in \{1, \ldots, l\}$ are leaving or joining nodes. The number L of joining or leaving nodes is randomly distributed. Assuming $L \sim \operatorname{Bin}(|\mathcal{N}|, p_{\Delta\mathcal{N}})$, where $|\mathcal{N}|$ is the number of nodes in the network, leads to the PMF

$$f_{\operatorname{rk}(\Delta \boldsymbol{A}_i)}(\tau) \leq f_{\overline{\operatorname{rk}(\Delta \boldsymbol{A}_i)}}(\tau) = \sum_{\ell=0}^{|\mathcal{N}|} f_L(\ell) \cdot f_W^{(*)\ell}(\tau). \tag{2.19}$$

Here $f_L(\ell)$ is the PMF of the random variable L, i.e. the number of joining or leaving nodes and $f_w(\tau)$ is the PMF of the random variable W corresponding to the node weight w, which can be upper bounded by a binomial distribution with parameters $|\mathcal{N}|$ and p_w. Further $f_W^{(*)\ell}(\tau)$ is the ℓ-fold convolution of $f_W(\tau)$ with itself. In Figure 2.5 the PMF is shown for values, that were chosen in the simulations in Section 6.4.2. The parameters are explained separately in Table 2.3.

2.4.2. Decoding Guarantee

As derived in [SCFH13], the end-to-end channel resembles an AMC with an effective error matrix $\boldsymbol{E}_i$. The rank of $\boldsymbol{E}_i$ is then crucial for the question whether the correct codeword in the DLNC scenario can be retrieved from the receive matrix $\boldsymbol{Y}_i$. Note that DLNC is a modulation scheme, i.e. the error correction capability originates from the Gabidulin code, that is applied on each information word. Hence we refer to the respective Gabidulin code, when considering code properties of DLNC.

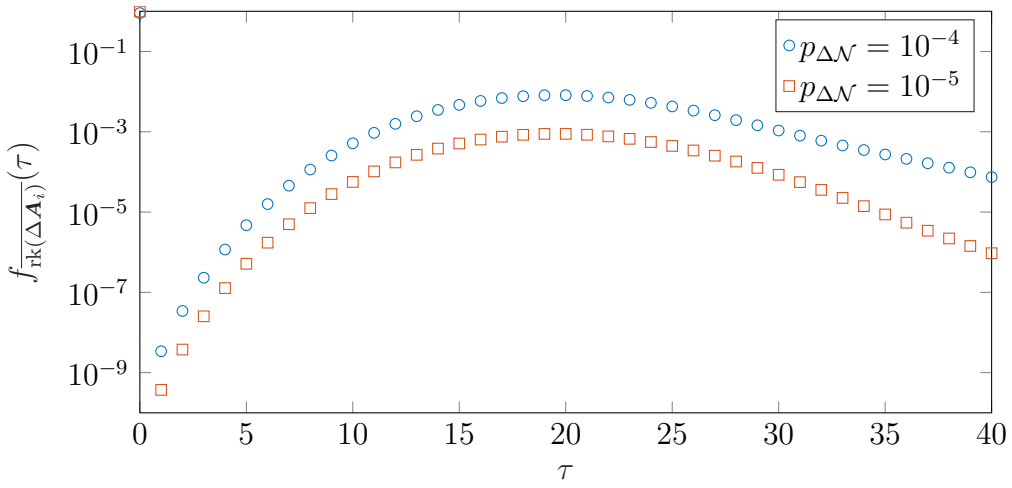


Figure 2.5.: PMF of the upper bound of the rank of the channel deviation for parameters as used in the simulations in Section 6.4.2, namely $|\mathcal{N}| = 1000$, $p_w = 0.01$, $p_{\Delta\mathcal{N}}$ as given in the plot.

Parameters	Definitions
$\lvert\mathcal{N}\rvert$	Number of nodes in the network
$p_{\Delta\mathcal{N}}$	Probability that a node leaves or joins the network during one generation.
p_w	Interconnection probability, i.e. the probability that there is an edge between two distinct nodes.

Table 2.3.: Definitions of the parameters for varying networks, all assumptions stick to [PCF+15].

Theorem 2.4.3 [SCFH13, PCF+15]
Let $\boldsymbol{Y}_i = \boldsymbol{A}_i\boldsymbol{X}_i + \boldsymbol{B}_i$ describe a RLNC scenario, where $\boldsymbol{X}_i$ is a with information matrix $\boldsymbol{S}_i$ DLNC modulated transmit matrix. More precisely $\boldsymbol{X}_i = (\boldsymbol{X}_{i-1})_{[n]} \cdot \boldsymbol{S}_i$, cf. (2.15), where $\boldsymbol{X}_{i-1}$ is the preceding transmit matrix. By DLNC demodulation $\hat{\boldsymbol{S}}_i$ is obtained as in (2.18), i.e. $\hat{\boldsymbol{S}}_i = (\boldsymbol{Y}_{i-1})^+_{[n]} \cdot \boldsymbol{Y}_i$, with $(\boldsymbol{Y}_{i-1})^+_{[n]}$ fulfilling (2.17). Then the effective error matrix calculates to

$$\boldsymbol{E}_i = \hat{\boldsymbol{S}}_i - \boldsymbol{S}_i = \boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\boldsymbol{S}_i + (\boldsymbol{Y}_{i-1})^+_{[n]}\boldsymbol{B}_{i-1}\boldsymbol{S}_i + (\boldsymbol{Y}_{i-1})^+_{[n]}\boldsymbol{B}_i + (\boldsymbol{Y}_{i-1})^+_{[n]}\Delta\boldsymbol{A}_i\boldsymbol{X}_i.$$

Let further the applied Gabidulin code have minimum distance d_{rk}. Then decoding is possible if and only if

$$\mathrm{rk}\,(\boldsymbol{E}_i) \leq \left\lfloor \frac{d_{\mathrm{rk}} - 1}{2} \right\rfloor.$$

Since $\boldsymbol{S}_i$, $(\boldsymbol{Y}_{i-1})^+_{[n]}$ and $\boldsymbol{X}_i$ have full rank n, an upper bound for its rank can be given by applying (2.3).

Corollary 2.4.4 [PCF$^+$15, Theorem 2]
The effective error matrix as given in Theorem 2.4.3 can be approximated by

$$\operatorname{rk}(\boldsymbol{E}_i) \leq \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) + \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) + \operatorname{rk}(\Delta\boldsymbol{A}_i)\,. \tag{2.20}$$

Consequently the error correction is possible if

$$\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) + \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) + \operatorname{rk}(\Delta\boldsymbol{A}_i) \leq \left\lfloor \frac{d_{\mathrm{rk}} - 1}{2} \right\rfloor,$$

and a Gabidulin code with minimum distance d_{rk} is applied.

3. Preparation

In this chapter we discuss the impact of the full-rank restriction for codewords in DLNC on the cardinality as well as the rate. Furthermore, lifted interleaved Gabidulin codes are considered in combination with lifting.

3.1. Full-Rank Restriction

For DLNC it is necessary to claim that the front part of the transmit matrices has full rank. Therefore we are interested in the proportion of regular matrices under all $n \times n$-matrices in $\mathbb{F}_q$. Another reason is that the Probability Density Function (PDF) of the rank of channel matrix $\boldsymbol{A}$ is helpful e.g. for Section 3.1.1.

We denote the number of Matrices $\in \mathbb{F}_q^{m \times n}$ of rank r as $M_r(n, m, q)$. If $m \geq n$ i.e. if the matrix is tall or square, the number is calculated as follows

$$M_r(n, m, q) = \begin{bmatrix} m \\ r \end{bmatrix}_q \prod_{i=0}^{r-1} (q^n - q^i). \tag{3.1}$$

On the other hand, for fat matrices ($n \times m$-matrix, where $m \geq n$), the number is calculated by

$$M_r(n, m, q) = \begin{bmatrix} m \\ r \end{bmatrix}_q \sum_{i=0}^{r} (-1)^{r-i} \begin{bmatrix} r \\ i \end{bmatrix}_q q^{ni + \binom{r-i}{2}}.$$

As indicated, for square matrices either formula can be used. The formulas were found in [vLW01] in Chapter 25. Hereby $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}_q$ is the Gaussian coefficient [KK08], also called Gaussian polynomial or q-binomial coefficient. It is q-analog to the binomial coefficient and defined as

$$\begin{bmatrix} n \\ \ell \end{bmatrix}_q = \prod_{j=0}^{\ell-1} \frac{(q^n - q^j)}{(q^\ell - q^j)} = \prod_{j=0}^{\ell-1} \frac{(1 - q^{n-j})}{(1 - q^{j+1})} \qquad \forall \ell \leq n.$$

In case of $\ell > n$ the Gaussian coefficient is defined to be zero. The binomial coefficient is obtained for $q \to 1^-$, since

$$\lim_{q \to 1^-} \frac{1-q^\alpha}{1-q} = \alpha.$$

For more information on q-analogs see [KS98]. The probability, that from the set of all $n \times n$-matrices a singular matrix is drawn, accordingly calculates to

$$\Pr\{\mathrm{rk}(\boldsymbol{A}) = r\} = \frac{1}{q^{n^2}} \begin{bmatrix} n \\ r \end{bmatrix}_q \sum_{i=0}^{r} (-1)^{r-i} \begin{bmatrix} r \\ i \end{bmatrix}_q q^{ni+\binom{r-i}{2}} = \frac{1}{q^{n^2}} \begin{bmatrix} n \\ r \end{bmatrix}_q \prod_{i=0}^{r-1} (q^n - q^i). \quad (3.2)$$

Cyran showed in [Cyr17] that for $q > 4$ one can derive (referring to equation (3.69))

$$\Pr\{\mathrm{rk}(\boldsymbol{A}) < n\} \approx \frac{1}{q}. \quad (3.3)$$

Note that this tends to zero for growing q. We will use this result later.

3.1.1. Application of the Rank Distribution

Regard the beforehand introduced matrix $\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top$, which is used in DLNC. It is useful to know the statistics of this matrix (cf. end of Chapter 4). Since $\boldsymbol{Y}_{i-1}^+$ and $\boldsymbol{X}_i$ have full rank, one can upper bound $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top\right)$ as follows

$$\begin{aligned} \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top\right) &= n - \mathrm{rk}\left(\boldsymbol{Y}_{i-1}^+ \cdot \boldsymbol{Y}_i\right) = n - \mathrm{rk}\left(\boldsymbol{Y}_i\right) && (3.4) \\ &= n - \underbrace{\mathrm{rk}\left(\boldsymbol{A}_i\boldsymbol{X}_i + \boldsymbol{B}_i\right)}_{\geq |\mathrm{rk}(\boldsymbol{A}_i\boldsymbol{X}_i) - \mathrm{rk}(\boldsymbol{B}_i)|} \\ &\overset{(2.4)}{\leq} n - \mathrm{rk}\left(\boldsymbol{A}_i\boldsymbol{X}_i\right) + \mathrm{rk}\left(\boldsymbol{B}_i\right) \\ &= n - \mathrm{rk}\left(\boldsymbol{A}_i\right) + \mathrm{rk}\left(\boldsymbol{B}_i\right). && (3.5) \end{aligned}$$

For the bound we use the assumption that $\mathrm{rk}(\boldsymbol{A}_i) \geq \mathrm{rk}(\boldsymbol{B}_i)$, which is true with high probability. Due to (3.4), we call $\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top$ the Receive Rank Deficiency Matrix (RRDM). The PDF of $\mathrm{rk}(\boldsymbol{A}_i)$ can be expressed by the fraction of the number of matrices of a certain rank versus the number of all $n \times n$-matrices as given in (3.2). As mentioned before the probability for full-rank matrices is quite high for large q. From (3.3) we can conclude, that the probability $\Pr\{\mathrm{rk}(\boldsymbol{A}) = n\} \approx 1 - \frac{1}{q}$ is close to one for high q and therefore contributes most to the PDF of $\mathrm{rk}(\boldsymbol{A})$. A lower bound for the

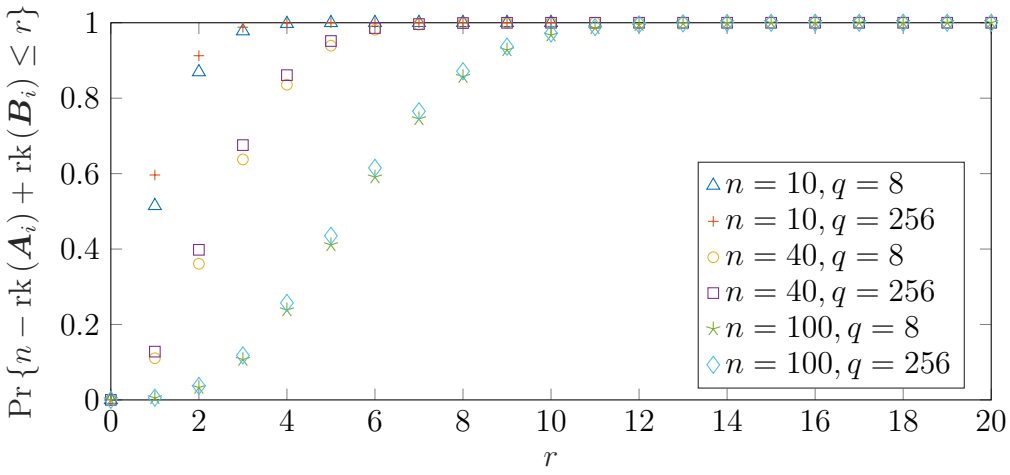


Figure 3.1.: A lower bound for the CDF of $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ for several q and n, where $p_{\mathrm{B}} = 0.05$.

Cumulative Density Function (CDF) of $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ can be given by

$$\begin{aligned}\Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq r\right\} &\geq \Pr\left\{n - \mathrm{rk}\left(\boldsymbol{A}_i\right) + \mathrm{rk}\left(\boldsymbol{B}_i\right) \leq r\right\} \\ &= \sum_{i=0}^{n} \Pr\left\{\mathrm{rk}\left(\boldsymbol{B}_i\right) = i\right\} \Pr\left\{n - \mathrm{rk}\left(\boldsymbol{A}_i\right) + i \leq r\right\} \\ &= \sum_{i=0}^{n} \Pr\left\{\mathrm{rk}\left(\boldsymbol{B}_i\right) = i\right\} \Pr\left\{\mathrm{rk}\left(\boldsymbol{A}_i\right) \geq n - r + i\right\} \\ &= \sum_{i=0}^{n} \Pr\left\{\mathrm{rk}\left(\boldsymbol{B}_i\right) = i\right\} \left(1 - \Pr\left\{\mathrm{rk}\left(\boldsymbol{A}_i\right) \leq n - r + i\right\}\right),\end{aligned} \tag{3.6}$$

using the law of total probability. These components can be calculated. The examples in Figure 3.1 show, that we can assume the probability $\Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq r\right\}$ to be quite high, even if r is relatively small.

3.1.2. Considerations on Full-Rank Gabidulin Codewords

So far we have considered uncoded matrices. Now it is time to turn to rank-metric codes. At first we consider square matrices, i.e. $m = n$. According to Gabidulin [Gab85, eq. (18)], the number of codes of rank weight n of MRD codes is calculated by

$$A_n(n, m, q, d) = \sum_{i=0}^{n-d} (-1)^{i+n-d} \begin{bmatrix} n \\ d+i \end{bmatrix}_q q^{(n-d-i)(n-d-i-1)/2} (q^{m(i+1)} - 1). \tag{3.7}$$

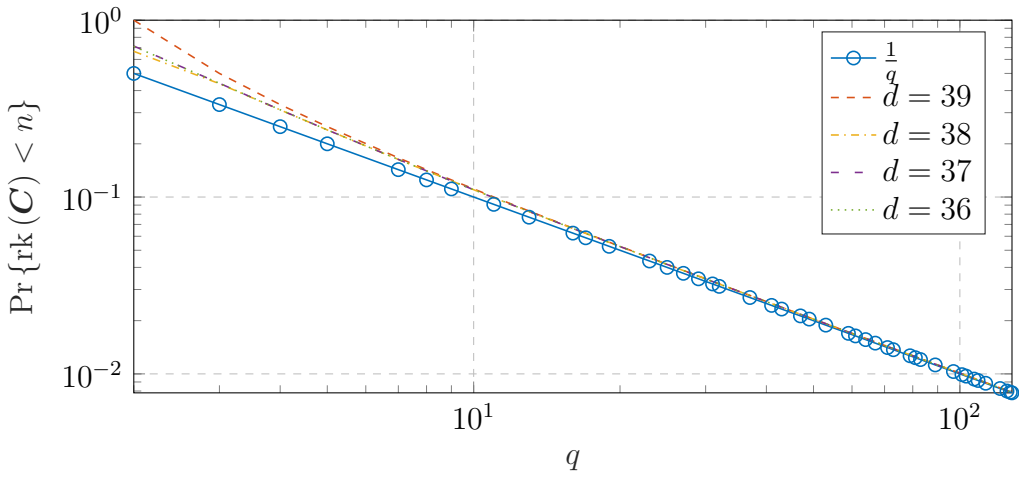


Figure 3.2.: Relative number of singular square Gabidulin codeword matrices of $\mathcal{G}[n, n-d+1]$ with $n = m = 40$ and minimum distance d calculated via (3.8). For $d < n-3$ the relative number of singular Gabidulin codewords does not vary cosiderably from the case $d = n-3$.

Therefore the probability, that a randomly chosen Gabidulin codeword does not have full rank, can be calculated. Let $\boldsymbol{C} \in \mathbb{F}_q^{m\times n}$ be a Gabidulin codeword, then

$$\Pr\{\mathrm{rk}(\boldsymbol{C}) < n\} = 1 - \frac{A_n(n,m,q,d)}{q^{nk}}. \tag{3.8}$$

Calculating the number $A_n(n,m,q,d)$ for several values shows, that the number of rank-deficient, i.e. singular matrices in the Gabidulin code behaves similarly to singular $n \times n$-matrices. Consequently, not being able to choose singular matrices as information in DLNC is quite close to choosing only full-rank Gabidulin codewords. Variations in the code rate were examined by changing k, whereas n stayed the same. The variation affects the behavior of $A_n(n,m,q,d)$ only slightly, as seen in Figure 3.2, except for $k = 1$ where the $A_n(n,m,q,d) = (q^m - 1)$ and the loss therefore calculates to $1/q^m$. For $k \geq 4 \Leftrightarrow d \leq n-3$ the curves for $\Pr\{\mathrm{rk}(\boldsymbol{C}) < n\}$ are very close to the case $d = n-3$. As long as $n \geq 10$ the curves hardly vary for different n.

For non-square Gabidulin codewords we have

$$\Pr\{\mathrm{rk}(\boldsymbol{C}) < n\} = 1 - \frac{A_n(n,m,q,d)}{q^{mk}}. \tag{3.9}$$

Since the number of overall codewords is affected more than the number of full-rank Gabidulin codewords (regard the term $(q^{m(i+1)} - 1)$ in the sum), the loss is even smaller for $m > n$. Nevertheless we are restricted to codewords that fulfill the full-rank condition in the first n columns. Simulations show that this constraint also

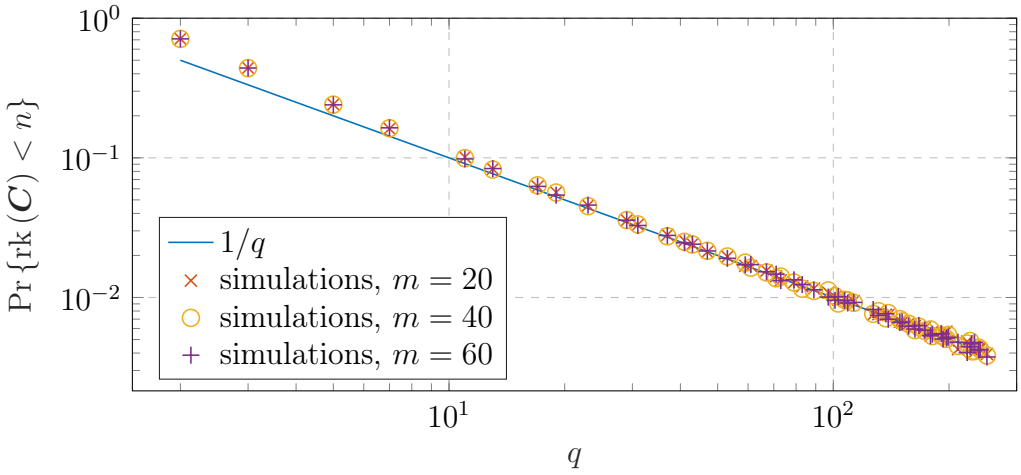


Figure 3.3.: Simulations for the relative number of non-square full-rank Gabidulin codewords of a Gabidulin code over $\mathbb{F}_{q^m}$ of length $n = 10$ and dimension $k = 8$. For the simulations 10000 random transposed Gabidulin codeword matrices were tested for their rank in the first n columns.

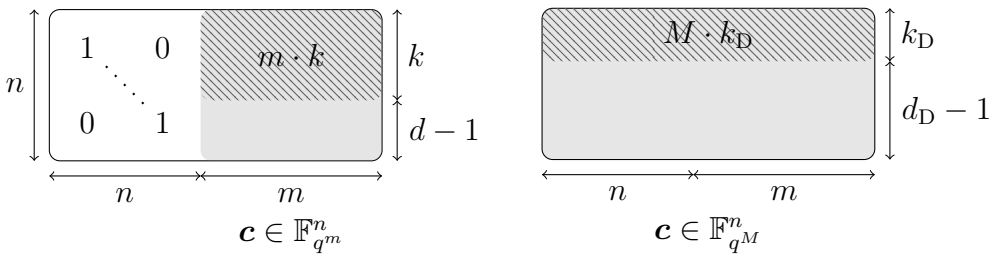
leads to a loss of $1/q$, see Figure 3.3. So the statistics for uncoded matrices is very close to the one of Gabidulin encoded matrices and especially to the statistics of square matrices, thus we can assume this loss in the coded case as well.

3.2. Rate Considerations

In order to compare the lifting construction and DLNC on a fair basis, the overall rate R of both schemes must be equal. The parameters n, m and q shall be the same for both methods. The code used for lifting is defined over $\mathbb{F}_{q^m}$, while the one for DLNC uses $\mathbb{F}_{q^M}$, where $M = n + m$. Given this condition, the dimension k_D can be chosen lower than the one for the lifting construction. Therefore DLNC has the advantage of being able to correct more errors than the lifted Gabidulin code. This can be seen in Figure 3.4. The code rate for Gabidulin codes over $\mathbb{F}_q^M$ is calculated as usual by

$$r = \frac{\log_q(\mathcal{M})}{Mn} \underline{\underline{\mathcal{M}=q^{Mk}}} \frac{k}{n}. \tag{3.10}$$

Yet regarding the matrices in Fig. 3.4, reveals that the identity matrix of the front part of the lifted Gabidulin codeword contributes to the number of sent symbols and therefore must be considered. Thus the overall rate for a lifted Gabidulin code



(a) Lifted Gabidulin Codeword (b) DLNC

Figure 3.4.: Comparison of lifted Gabidulin and usual Gabidulin codewords, where the matrices are of same size. The shaded part indicates the information symbols. The whole codeword is highlighted in gray. Note that $M = n + m$.

$\mathcal{G}_{\mathrm{L}}[n, k_{\mathrm{L}}]$ over $\mathbb{F}_{q^m}$ calculates as

$$R_{\mathrm{L}} = \frac{m}{n+m}\frac{k_{\mathrm{L}}}{n}. \tag{3.11}$$

Important to notice is, that it can never be greater than $\frac{m}{n+m}$, cf.

$$R_{\mathrm{L}} = \frac{m}{n+m}\frac{k_{\mathrm{L}}}{n} \overset{k_{\mathrm{L}} \leq n}{\leq} \frac{m}{n+m} =: R_{\max}, \tag{3.12}$$

i.e. for higher rates, the procedures DLNC and lifting are not comparable. Certainly for $m \to \infty$ the fraction approaches 1. But choosing m too high is not a feasible solution, since the computational complexity grows with the field size, cf. [SKK08, Appendix E], [Wac13, Section 3.1.1]. The overall rate for the Gabidulin code $\mathcal{G}[n, k_{\mathrm{D}}]$ used in DLNC defined over $\mathbb{F}_{q^M}$ reads

$$R_{\mathrm{D}} = \frac{k_{\mathrm{D}}}{n}\left(1 - \frac{n}{qM}\right), \tag{3.13}$$

when using the loss introduced in [SCFH13]. As one can see the higher q and the higher m (compared to n), the closer the overall rate gets to the code rate. Although the approximation of $1/q$ could not analytically be derived for full-rank non-square Gabidulin codewords, we conjecture this loss in the number of matrices. With this assumption we have a code cardinality of

$$\mathcal{M}_{\mathrm{D}} = q^{Mk_{\mathrm{D}}}\left(1 - \frac{1}{q}\right) \tag{3.14}$$

for DLNC. Therefore we obtain as overall rate of symbols

$$R_{\mathcal{M}_{\mathrm{D}}} = \frac{\log_q(\mathcal{M}_{\mathrm{D}})}{Mn} \overset{(3.14)}{=} \frac{Mk_{\mathrm{D}} + \log_q\left(1 - \frac{1}{q}\right)}{Mn} = \frac{k_{\mathrm{D}}}{n} - \frac{1}{Mn}\log_q\left(\frac{q}{q-1}\right). \tag{3.15}$$

q	k_D	$r = \frac{k_\mathrm{D}}{n}$	M	R_D	$R_{\mathcal{M}_\mathrm{D}}$
16	18	$\frac{18}{40} = 0.45$	80	$\frac{279}{640} \approx 0.4359$	≈ 0.449993
			120	$\frac{141}{320} \approx 0.4406$	≈ 0.449995
	24	$\frac{24}{40} = 0.6$	80	$\frac{93}{160} \approx 0.5813$	≈ 0.599993
			120	$\frac{47}{80} = 0.5875$	≈ 0.599995
256	18	$\frac{18}{40} = 0.45$	80	$\frac{4599}{10240} \approx 0.4491$	≈ 0.44999998
			120	$\frac{2301}{5120} \approx 0.4494$	≈ 0.44999999
	24	$\frac{24}{40} = 0.6$	80	$\frac{1533}{2560} \approx 0.5988$	≈ 0.5999998
			120	$\frac{767}{1260} \approx 0.5992$	≈ 0.5999999

Table 3.1.: Comparison of examples of the overall rates R_D from equation (3.13) and $R_{\mathcal{M}_\mathrm{D}}$, equation (3.15), for $n = 40$.

In Table 3.1 the two definitions of overall rates are compared for several parameter values. One can see that for high q and M, the rates do not differ considerably. Nevertheless the derived overall rate $R_{\mathcal{M}_\mathrm{D}}$ from (3.15) approaches the code rate much faster. Since the rate loss L_DLNC introduced by [SCFH13] is more conservative, we will stick to it. Note that regarding the relative number of matrices, this loss is a rather bad approximation. For equal overall rates the dependency between the dimensions is derived from

$$\begin{aligned} \frac{mk_\mathrm{L}}{Mn} &= \frac{k_\mathrm{D}}{n}\left(1 - \frac{n}{qM}\right) \\ \iff \qquad k_\mathrm{L} &= k_\mathrm{D}\frac{M}{m}\underbrace{\left(1 - \frac{n}{qM}\right)}_{\to 1 \text{ for } q\to\infty}. \end{aligned} \tag{3.16}$$

The dimensions must be integers, therefore rounding might be necessary, which complicates the search for suitable parameters for equal overall rates. Therefore we allow small variations between the rates. Since $M > m$, we have $k_\mathrm{D} < k_\mathrm{L}$. DLNC therefore has a higher minimum distance than lifting. However, the better error correction capability is necessary, for the DLNC case has generally to cope with a higher error rank than the lifting case, due to more terms contributing to the rank. For a certain factor δ between the error correction radii one can derive the following code rate region.

Theorem 3.2.1
Let τ_{D} be the error correction radius of a Gabidulin code $\mathcal{G}_{\mathrm{D}}[n, k_{\mathrm{D}}]$ over $\mathbb{F}_{q^M}$ used for DLNC and τ_{L} the error correction radius of a lifted Gabidulin code $\mathcal{G}[n, k_{\mathrm{L}}]$ over $\mathbb{F}_{q^m}$, where $M = n+m$. Further let $\delta > 1$ and minimum distance $d_{\mathrm{rk,D}} = n - k_{\mathrm{D}} + 1$ odd and the overall rates of both schemes be equal, i.e. (3.16) *holds. Then*

$$\delta\tau_{\mathrm{L}} \leq \tau_{\mathrm{D}} \iff \frac{k_{\mathrm{L}}}{n} \geq \frac{\delta - 1}{\delta - \frac{m}{M\left(1-\frac{n}{qM}\right)}}. \tag{3.17}$$

Proof:

$$\begin{aligned}
& & \delta\tau_{\mathrm{L}} \leq & \tau_{\mathrm{D}} \\
\overset{\tau=\lfloor (d_{\mathrm{rk}}-1)/2 \rfloor}{\Longleftrightarrow} & & \delta\frac{d_{\mathrm{rk,L}}-1}{2} \leq & \frac{d_{\mathrm{rk,D}}-1}{2} \qquad (3.18) \\
\overset{(3.16)}{\Longleftrightarrow} & & \delta(n-k_{\mathrm{L}}) \leq & n - k_{\mathrm{L}}\frac{m}{M}\left(1-\frac{n}{qM}\right)^{-1} \\
\Longleftrightarrow & & (\delta-1)n \leq & \left(\delta - \frac{m}{M}\left(1-\frac{n}{qM}\right)^{-1}\right) k_{\mathrm{L}} \\
\Longleftrightarrow & & \frac{k_{\mathrm{L}}}{n} \geq & \frac{\delta-1}{\delta - \frac{m}{M\left(1-\frac{n}{qM}\right)}}. \qquad (3.19)
\end{aligned}$$

Note that the rounding in (3.18) can be neglected if and only if we demand $d_{\mathrm{rk,D}}$ to be odd. □

This scenario could be referred to as an adversary channel, since this statement is most useful if the channel always introduces the maximum amount of τ_{L} (for lifting) respectively τ_{D} (for DLNC) errors. For the probabilistic analysis it is nevertheless reasonable, as it provides a lower bound for the code rate k_{L}/n, where the allegations made in further chapters hold. Since we assume $\delta\tau_{\mathrm{L}} = \tau_{\mathrm{D}}$ in these chapters, the statement must be slightly changed.

Lemma 3.2.2
Let τ_{D} be the error correction radius of a Gabidulin code $\mathcal{G}_{\mathrm{D}}[n, k_{\mathrm{D}}]$ over $\mathbb{F}_{q^M}$ used for DLNC and τ_{L} the error correction radius of a lifted Gabidulin code $\mathcal{G}[n, k_{\mathrm{L}}]$ over $\mathbb{F}_{q^m}$, where $M = n + m$. Further let $\delta > 1$, minimum distance $d_{\mathrm{rk,D}}$ and $d_{\mathrm{rk,L}}$ both odd and $R_{\mathrm{D}} \geq R_{\mathrm{L}}$. Then

$$\delta\tau_{\mathrm{L}} = \tau_{\mathrm{D}} \iff \frac{k_{\mathrm{L}}}{n} \geq \frac{\delta - 1}{\delta - \frac{m}{M\left(1-\frac{n}{qM}\right)}}. \tag{3.20}$$

Proof:

$$\begin{aligned}
& \frac{k_\mathrm{D}}{n}\left(1-\frac{n}{qM}\right) \geq \frac{m}{M}\frac{k_\mathrm{L}}{n} \\
\iff \quad & \delta k_\mathrm{L} - (\delta-1)n \geq \frac{mk_\mathrm{L}}{M}\left(1-\frac{n}{qM}\right)^{-1} \\
\iff \quad & k_\mathrm{L}\left(\delta - \frac{m}{M}\left(1-\frac{n}{qM}\right)^{-1}\right) \geq (\delta-1)n \\
\iff \quad & \frac{k_\mathrm{L}}{n} \geq \frac{\delta-1}{\delta - \frac{m}{M}\left(1-\frac{n}{qM}\right)^{-1}},
\end{aligned}$$

where we get $k_\mathrm{D} = \delta k_\mathrm{L} - (\delta-1)n$ from

$$\delta\frac{n-k_\mathrm{L}}{2} = \delta\tau_\mathrm{L} = \tau_\mathrm{D} = \frac{n-k_\mathrm{D}}{2}.$$

□

If q is chosen high enough, the loss due to the full-rank codeword constraint can be disregarded as indicated in equation (3.16). Then the lower bound for the code rate depends (apart from δ) on the term $\frac{m}{M}\left(1-\frac{n}{qM}\right)^{-1}$, i.e. on n and m. Later we will regard theorems, where we find out, which code length one has to choose at least, such that DLNC has a lower failure probability than lifting. Therefore one might not want to fix n in the beginning. This is why we introduce a factor f, that replaces the dependency of the lower bound of the code rate on q, n and m. Let

$$\frac{m}{M}\left(1-\frac{n}{qM}\right)^{-1} = \frac{m}{m+n\frac{q-1}{q}} =: \frac{1}{f}. \tag{3.21}$$

For large q this resembles the relation of m to $M = n+m$. Instead of fixing m and n in the beginning, one can hence set f as desired and later adjust m for the code length n, obtained in Chapter 4. Due to $0 < n \leq m$ and $q > 1$ it is $f \in (1,2)$. Note that for $m = n$ it is

$$f \xrightarrow{q\to\infty} 2.$$

We define

$$r_{\min,f} := \frac{\delta-1}{\delta-\frac{1}{f}}. \tag{3.22}$$

Using $f = 2$ yields the smallest bound for the code rate, which will be used for comparison later on. Note that this is the worst case for lifting. With this simplified case where $m = n$ and $q \to \infty$, i.e. $R_{\mathrm{D}} = k_{\mathrm{D}}/n$, $R_{\mathrm{L}} = k_{\mathrm{L}}/(2n)$ (for equal overall rates we therefore have $k_{\mathrm{D}} = \frac{1}{2}k_{\mathrm{L}}$), we get

$$\frac{k_{\mathrm{L}}}{n} \overset{(3.22)}{\geq} \frac{\delta - 1}{\delta - 1/2} =: r_{\min,2} \tag{3.23}$$

$$\implies \qquad R_{\mathrm{L}} = R_{\mathrm{D}} \geq \frac{1}{2}\frac{\delta - 1}{\delta - 1/2} = \frac{\delta - 1}{2\delta - 1}. \tag{3.24}$$

This implies, that e.g. for the case $\delta = 2$, which is the intuitive view on the problem, one can say that DLNC supersedes lifting for overall rates greater 1/3.

3.3. Considerations on Lifted Interleaved Gabidulin Codes

Interleaved Gabidulin codes are obtained by aligning several Gabidulin codewords vertically or horizontally. We refer to the vertical concatenation and restrict to homogeneous interleaved codes, where all concatenated codes have the same dimension, for simplicity. The definition uses the notation introduced in [Wac13, Def. 2.17].

Definition 3.3.1 (Homogeneous Interleaved Gabidulin codes [Wac13])
Let $g_0, g_1, \ldots, g_{n-1} \in \mathbb{F}_{q^m}$ be fixed elements and linearly independent over $\mathbb{F}_q$. A linear vertically homogeneous interleaved Gabidulin code $\mathcal{G}[n, k, L]$ over $\mathbb{F}_{q^m}$ of length $n \leq m$ and dimension $k \leq n$ is then the following set

$$\mathcal{G}[n,k,L] \coloneqq \left\{ \begin{pmatrix} f^{(1)}(g_0) & f^{(1)}(g_1) & \ldots & f^{(1)}(g_n) \\ f^{(2)}(g_0) & f^{(2)}(g_1) & \ldots & f^{(2)}(g_n) \\ \vdots & \vdots & & \vdots \\ f^{(L)}(g_0) & f^{(L)}(g_1) & \ldots & f^{(L)}(g_n) \end{pmatrix} : f(x) \in \mathbb{L}_{q^m}[x], \right.$$
$$\left. \deg_q f^{(i)}(x) < k \ \forall i \in \{1, \ldots, L\} \right\}.$$

L is called interleaving order.

These codes represent the equivalent of interleaved RS codes in rank metric. Interleaving establishes the possibility to decode beyong half the minimum distance d_{rk} of the code.

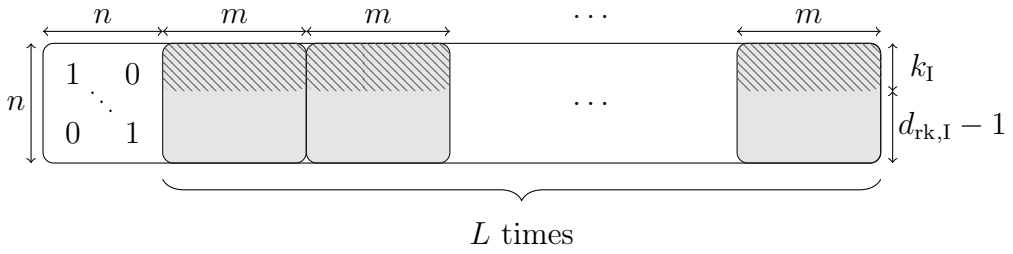


Figure 3.5.: Codeword of a lifted interleaved Gabidulin code $\mathcal{G}[n, k_{\mathrm{I}}, L]$ over $\mathbb{F}_{q^m}$ of length n and interleaving order L, where the interleaved Gabidulin codeword is transposed before lifting. The gray parts indicate the concatenated codewords $(f^{(i)}(g_0), f^{(i)}(g_1), \ldots, f^{(i)}(g_n))$ of the underlying Gabidulin code.

Theorem 3.3.2 [Ove07, Theorem 3.9]
Let $\mathcal{G}[n, k, L]$ be an interleaved Gabidulin code. Then for error rank

$$\epsilon \leq \frac{L}{L+1}(n-k) \tag{3.25}$$

the probability that a receive word is decoded falsely is upper bounded by

$$\mathrm{Pr}_{\mathrm{fail}} \leq 1 - \left(1 - \frac{4}{q^m}\right)\left(\frac{q^{m\epsilon} - q^{mL}}{q^{m\epsilon}}\right),$$

as given in [Ove07, equation (12)] for $L \ll n - k$.

The success probability is high for reasonably large q and m. As one can see, for $L \to \infty$ one can approach the minimum distance as decoding bound. What must be considered, is that the statement in (3.25) only holds for $L \ll n-k = d_{\mathrm{rk}} - 1$, where d_{rk} is the minimum distance of the rank-metric code. This means that for high L we also need sufficiently large d_{rk}, i.e. for fixed n lower code rates are regarded. In order to use interleaved codes in RLNC the codewords are transposed and lifted. Figure 3.5 illustrates such a codeword.

The procedure of interleaving has the advantage, that the rate loss due to lifting is reduced, since the identity matrix is sent once for several codewords. Furthermore, since the computational expense grows with the field size, it is more efficient to decode several codewords in $\mathbb{F}_{q^m}$ than a respective non-interleaved Gabidulin codeword over $\mathbb{F}_{q^{Lm}}$ (compare [SKK08, Appendix E]). The problem with it is that the length of a packet is considerably longer, so that there might be buffer problems at inner nodes that have to store all incoming packets before combining them and the latency increases.

Most important, if the packet length is fixed, then the code length is relatively small compared to the packet size, i.e. $n \ll n + mL$. It is especially relatively smaller than

a lifted non-interleaved code, see

$$\frac{n}{n+mL} < \frac{n}{n+m} \text{ for } L > 1.$$

According to Shannon's work [Sha48], in a probabilistic channel the performance of a code is better the longer the code. Hence we can conclude that for a given packet length, the usage of interleaving might be disadvantageous. Nevertheless there are applications, where the packet size is not restricted, in such scenarios interleaving can improve the error correction capabilities without limiting the code length.

On the other hand, when using an interleaved Gabidulin code in a DLNC scheme, where we have the same packet size, the relative code length is larger than the one of the lifted code, due to the identity matrix of the lifting construction. Say we have interleaving length L for both schemes. Then the lifted interleaved codeword consists of $L+1$ submatrices, L of length m and one of length n. Since the interleaved Gabidulin code used for DLNC shall have the same interleaving order L, its code length n_{D} is upper bounded by

$$n_{\mathrm{D,max}} = m_{\mathrm{D}} = \frac{n+mL}{L} = \frac{n}{L} + m > m = n_{\max} \geq n.$$

This upper bound is larger than the one for the lifted Gabidulin code, where $n \leq m$. Therefore we can assume, that in a probabilistic channel the interleaved Gabidulin with DLNC scheme performs better than the lifted interleaved Gabidulin code. These considerations show that interleaved Gabidulin codes in combination with DLNC might do better compared to lifted interleaved Gabidulin codes in RLNC scenarios.

4. Static Networks

The word static means, that in this chapter the channel matrix $\boldsymbol{A}$ is assumed to stay the same between generations.

Definition 4.0.1 (Static Network [SCFH13])
Let $\boldsymbol{A}_i\boldsymbol{X}_i + \boldsymbol{B}_i$ define a RLNC network in generation i with channel matrix $\boldsymbol{A}_i$, transmit matrix $\boldsymbol{X}_i$ and error matrix $\boldsymbol{B}_i$. Then the network is called static if the channel matrix $\boldsymbol{A}_i$ is constant over all generations, i.e.

$$\Delta\boldsymbol{A}_i = \boldsymbol{A}_i - \boldsymbol{A}_{i-1} = \boldsymbol{0}.$$

The main goal of this work is to acquire knowledge about the nature of lifting and DLNC by probabilistic analysis. Therefore we need some preliminaries in this topic as well as others.

4.1. Preliminaries

Since we assume, that the rank of the error matrix is binomially distributed (cf. Section 2.3.2), we introduce some properties of the Binomial distribution here. Figure 4.1 displays some examples of the Binomial Distribution. A property, we make vast use of, is the following. The Binomial Distribution of the sum of two random variables $X_1 \sim \mathrm{Bin}(n_1, p)$ and $X_2 \sim \mathrm{Bin}(n_2, p)$ is calculated as follows. Let $\tau \in \mathbb{N}$, where $\tau \leq n_1 + n_2$, then

$$\Pr\{X_1 + X_2 \leq \tau\} = \sum_{k=0}^{\tau} \binom{n_1 + n_2}{k} p^k (1-p)^{n_1+n_2-k}.$$

In the case $n_1 = n_2 = n$ we therefore have

$$X_1, X_2 \sim \mathrm{Bin}(n, p) \Longrightarrow X_1 + X_2 \sim \mathrm{Bin}(2n, p). \tag{4.1}$$

In several upcoming proofs we need the concept of the Kullback-Leibler Divergence.

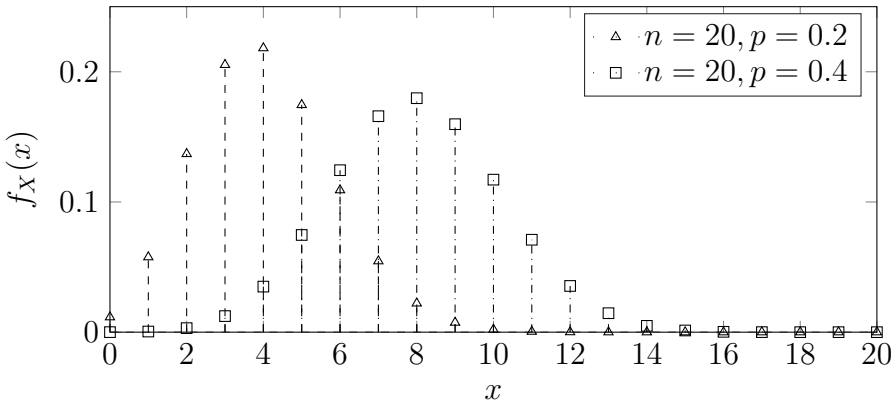


Figure 4.1.: Probability density function of Binomial Distributions, $X \sim \mathrm{Bin}\,(n, p)$. The shown parameters are not realistic for communication networks.

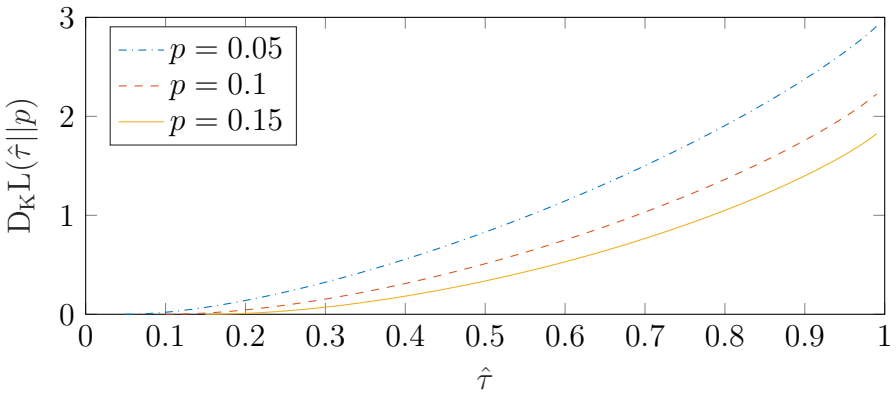


Figure 4.2.: Kullback-Leibler divergence for $\hat{\tau} \in (p, 1)$.

Definition 4.1.1 (Kullback-Leibler Divergence [KL51])
Let x and p be the success probabilities of $\mathrm{Ber}(x)$ *and* $\mathrm{Ber}(p)$ *distributed random variables. Then*

$$\mathrm{D_{KL}}(x||p) = x \ln \frac{x}{p} + (1 - x) \ln \frac{1 - x}{1 - p}$$

is called Kullback-Leibler divergence (or relative entropy) of two Bernoulli trials with probabilities x and p.

Figure 4.2 shows the Kullback-Leibler divergence for $\hat{\tau} > p$. The Kullback-Leibler divergence is used in the following two bounds for the binomial distribution.

Lemma 4.1.2 [Hoe63, Ash67]
Let $X \sim \mathrm{Bin}\,(n, p_\mathrm{B})$ *and* $\tau > np_\mathrm{B}$. *Then*

$$\Pr\{X \geq \tau\} \leq \exp\left\{-n\mathrm{D}_\mathrm{KL}\left(\frac{\tau}{n}\middle|\middle|p_\mathrm{B}\right)\right\} \qquad (<)$$

and

$$\Pr\{X \geq \tau\} \geq \frac{1}{\sqrt{2n}}\exp\left\{-n\mathrm{D}_\mathrm{KL}\left(\frac{\tau}{n}\middle|\middle|p_\mathrm{B}\right)\right\}. \qquad (>)$$

The first bound can be derived by the Chernoff bound and is known as Chernoff-Hoeffding Theorem. See [**?**] inequality (2.1) and recall, that the sum of Bernoulli-distributed random variables is binomially distributed. Combining inequality (4.7.2) from [Ash67] with $2^{-x} \geq \mathrm{e}^{-x}\ \forall x \geq 0$ and $x(1-x) \leq \frac{1}{4}$ for $p < x < 1$ leads to the second bound. Moreover we need the Lambert-W function (also called omega function or product function) introduced in [Lam58], i.e. the special case regarded in [Eul83], cf. also [CGH$^+$96].

Definition 4.1.3 (Lambert-W function [Lam58, Eul83])
*The Lambert-*W *function* $\mathrm{W} : \mathbb{C} \to \mathbb{C}$ *is defined to be the inverse of the function* $f(a) = a \cdot \mathrm{e}^a = z$, *meaning that*

$$\mathrm{W}(z) = a = f^{-1}(a \cdot \mathrm{e}^a) \qquad \forall a, z \in \mathbb{C}.$$

Note that it has several branches. This work is concerned with branches defined for real numbers. There are two real-valued branches, these are the main branches $\mathrm{W}_0(z) \geq -1$ and $\mathrm{W}_{-1}(z) \leq -1$. They are defined for $z \geq -\mathrm{e}^{-1}$, besides W_{-1} is only defined for non-positive numbers. Furthermore the function is two-valued for $z \in [-\mathrm{e}^{-1}, 0]$. Also important to note is that $\mathrm{W}_{-1}(z) \to -\infty$ for $z \to 0^-$. This is used later on. Another important property is that $\mathrm{W}_0(-\mathrm{e}^{-1}) = \mathrm{W}_{-1}(-\mathrm{e}^{-1}) = -1$. The two main branches of the Lambert-W function are depicted in Figure 4.3.

4.2. Simplified Comparison

The failure probabilities, to be considered, depend on the statistics of the rank of the error matrices. When considering the rank of an error matrix, we regard it as a random variable (RV). In the following we use these substitutions: $R_1 = \mathrm{rk}\,(\boldsymbol{B}_{i-1})$, $R_2 = \mathrm{rk}\,(\boldsymbol{B}_i)$ and $R_+ = \mathrm{rk}\,(\boldsymbol{B}_i) + \mathrm{rk}\,(\boldsymbol{B}_{i-1})$. Be aware that in the lifting case only one generation is regarded and therefore it is not necessary to distinguish between

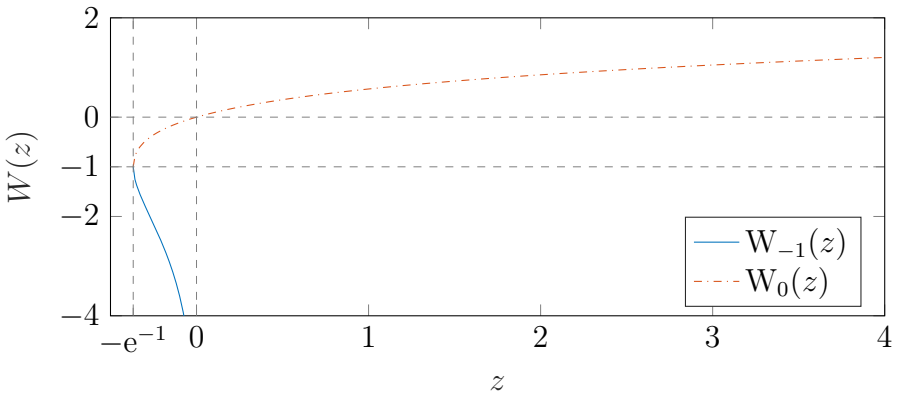


Figure 4.3.: Both main branches of Lambert-W function for real values z.

R_1 and R_2. Nevertheless both random variables are used in later proofs, which is why we introduced this notation.

At a first glance (with some neglections), DLNC has like DPSK on average to fight with twice the amount of errors, since the received matrix depends on two error matrices, i.e. the error rank is assumed to consist of the terms depicted in Figure 4.4.

$$\operatorname{rk}(\boldsymbol{E}_i) \leq \underbrace{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)}_{=0} + \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) + \underbrace{\operatorname{rk}(\Delta\boldsymbol{A}_i)}_{=0}$$

Figure 4.4.: Intuitive view on the error rank in DLNC, cf. equation (2.20), where $\operatorname{rk}(\Delta\boldsymbol{A}_i)$ and $\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ are assumed to be zero.

Therefore we will at first regard, how the sum of two equally distributed random variables behaves, compared to only one of it. In the best case for lifting it is possible to decode if $\operatorname{rk}(\boldsymbol{B}_i) \leq \tau = \lfloor (d_{\mathrm{rk,L}}-1)/2 \rfloor$, a decoding failure will consequently appear if the inequality is not fulfilled. In the case of DLNC, when we set $\operatorname{rk}(\Delta\boldsymbol{A}_i) = 0$ (due to the assumption of a static network) and $\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) = 0$ (since it is usually quite small) from (2.20), what stays is

$$\operatorname{rk}(\boldsymbol{E}_i) \leq \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) \leq \left\lfloor \frac{d_{\mathrm{rk,D}} - 1}{2} \right\rfloor = \tau_{\mathrm{D}}.$$

The failure probability is therefore determined by the probability that the sum of the rank of the error matrices $\boldsymbol{B}_i$ and $\boldsymbol{B}_{i-1}$ is greater than the respective correction

radius τ_{D}. Due to the naive assumption of the double amount of errors, we expect the error correction radius of DLNC $\tau_{\mathrm{D}} = 2\tau$ to be twice the size of the correction radius of the lifted Gabidulin code. All in all the question is under which conditions the statement

$$\Pr\{R_+ > 2\tau\} < \Pr\{R_2 > \tau\}$$

holds. In order to be able to apply the bounds ($<$) and ($>$) as they are on the probabilities we also include the error correction radius in the failure probabilities, i.e. we compare

$$\Pr\{R_+ \geq 2\tau\} < \Pr\{R_2 \geq \tau\}. \tag{4.2}$$

Only the case $\tau > \mathbb{E}[R_2] = np_{\mathrm{B}}$ (n and p_{B} defined as in Section 2.3.2) is regarded, as the number of correctable errors is always chosen larger than the expected number of errors. Otherwise the performance would be bad. Furthermore we define $\hat{\tau} \coloneqq \tau/n$ as normed version of the error correction radius τ of the lifted Gabidulin code. Note that there is a direct link between $\hat{\tau}$ and code rate r if MDS or MRD codes are used:

$$r = \frac{k}{n} = \frac{n-d+1}{n} = 1 - \frac{2\tau}{n} = 1 - 2\hat{\tau} \iff \hat{\tau} = \frac{1}{2}(1-r). \tag{4.3}$$

As a first step for the proof of the statement (4.2) above we introduce the following lemma.

Lemma 4.2.1
Let $\hat{\tau} > p_{\mathrm{B}}$ *and* $\mathrm{D}_{\mathrm{KL}}(\hat{\tau}||p_{\mathrm{B}}) \geq c > 0 \ \wedge \ c \leq \mathrm{e}^{-1}$. *If* $n > -\frac{1}{2c}\mathrm{W}_{-1}(-c)$, *where* $\mathrm{W}_{-1}(z)$ *is the* -1*-st branch of the Lambert-*W *function, then*

$$\exp\left\{-n\mathrm{D}_{\mathrm{KL}}\left(\hat{\tau}\middle||p_{\mathrm{B}}\right)\right\} < \frac{1}{\sqrt{2n}}.$$

Proof:
Since $n > -\frac{1}{2c}\mathrm{W}_{-1}(-c)$ we know that

$$\exists n_0 \leq n : n_0 = -\frac{1}{2c}\mathrm{W}_{-1}(-c), \tag{4.4}$$

where $n_0 \in \mathbb{R}$, since $c \leq \mathrm{e}^{-1}$ and $n_0 > 0$ due to $c > 0$. Equation (4.4) is equivalent to

$$\begin{aligned}
& \mathrm{W}_{-1}(-c) = -2cn_0 \\
\overset{\text{Def. W}}{\iff} \quad & -c = -2cn_0\mathrm{e}^{-2cn_0} \\
\iff \quad & \frac{1}{2n_0} = \mathrm{e}^{-2cn_0} \\
\iff \quad & 2n_0 = \mathrm{e}^{2cn_0},
\end{aligned}$$

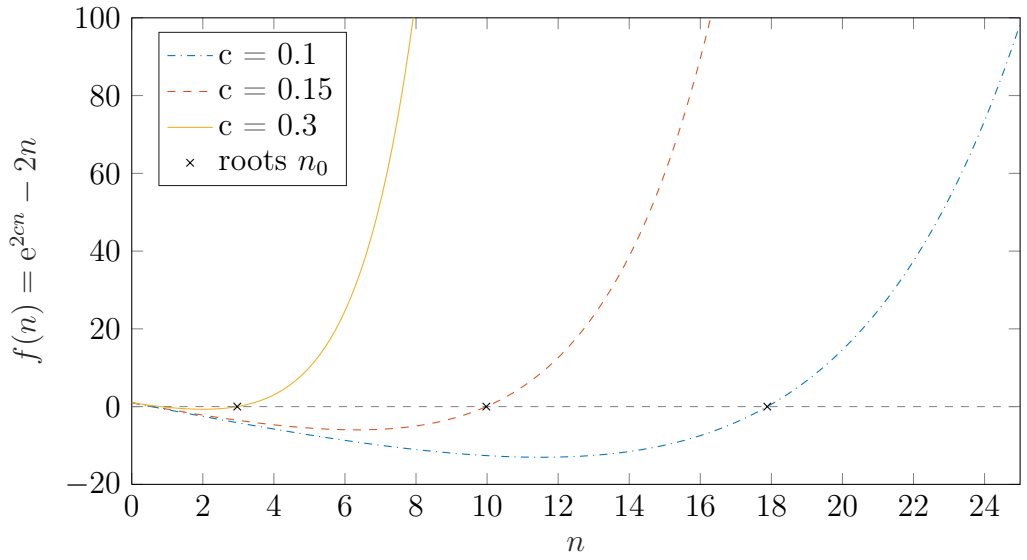


Figure 4.5.: Function $f(n)$ and its roots n_0 for several values of c.

implying that n_0 is a root of the function $f(n) = \mathrm{e}^{2cn} - 2n$. Its derivative calculates to

$$\frac{\mathrm{d}}{\mathrm{d}n} f(n) = 2c\mathrm{e}^{2cn} - 2 = 2(c\mathrm{e}^{2cn} - 1).$$

It is $f(0) = 1$ and $f'(0) = 2c - 2 \leq 2\mathrm{e}^{-1} - 2 < 0$ but also

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}n} f(n) > 0 \iff & \qquad \mathrm{e}^{2cn} > \frac{1}{c} \\ \iff & \qquad n > -\frac{1}{2c}\ln(c). \end{aligned}$$

Hence we have two roots $\in \mathbb{R}$, where the smaller one $\in (0, 1)$ is given by W_0 and the other one by W_{-1}. Defining n_1 by $\frac{\mathrm{d}}{\mathrm{d}n} f(n)|_{n=n_1} = 0$, i.e. the abscissa of the minimum, we have for $n_0 \geq n_1$:

$$\begin{aligned} n > n_0 = -\frac{1}{2c}\mathrm{W}_{-1}(-c) \implies & \qquad \mathrm{e}^{2cn} > 2n \\ \iff & \qquad \mathrm{e}^{-2cn} < \frac{1}{2n}. \end{aligned}$$

The function $f(n)$ and its root n_0 are illustrated for example values of c in Figure 4.5. Since the square root is strictly monotonically increasing, applying the square root

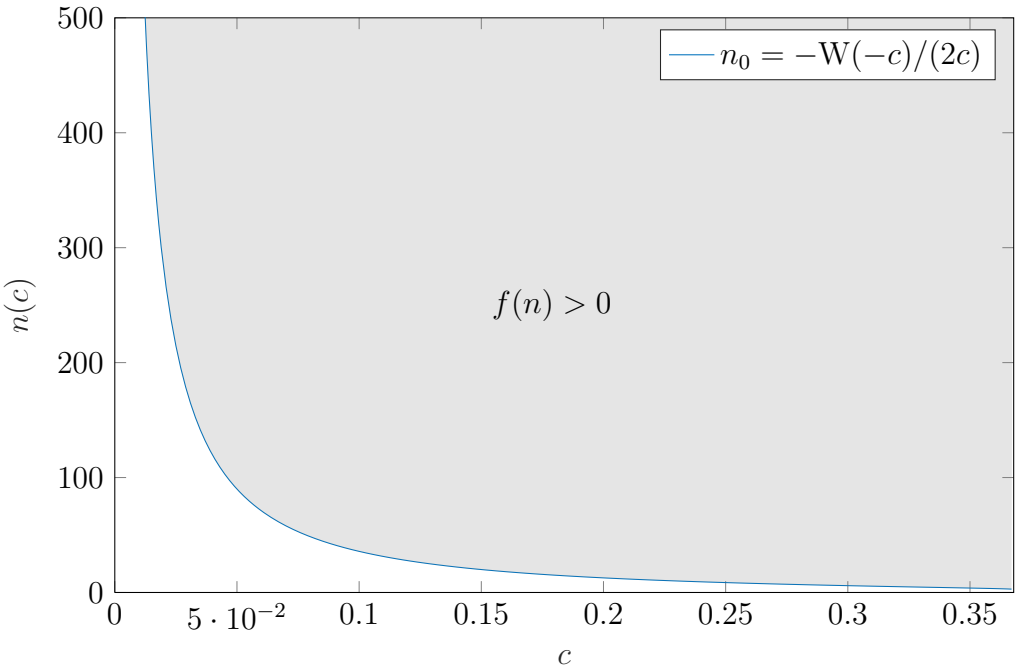


Figure 4.6.: Lower bound on $n : f(n) > 0$ dependent on c. The grey region fulfills the condition.

preserves the relation and therefore

$$\begin{aligned} & & \mathrm{e}^{-2cn} &< \frac{1}{2n} \\ & \xLeftrightarrow{c \leq \mathrm{D_{KL}}(\hat{\tau}||p_\mathrm{B})} & \mathrm{e}^{-n\mathrm{D_{KL}}(\hat{\tau}||p_\mathrm{B})} \leq \mathrm{e}^{-cn} &< \frac{1}{\sqrt{2n_0}}. \end{aligned}$$

□

The respective conditions on n_0 as a function of c can be found in Figure 4.6. Using this lemma, we can now state the first theorem concerning the failure probabilities. The main outcome of the theorem is that one has to choose n large enough for the failure probability of DLNC to be smaller than the one of lifted Gabidulin code. As one can see in Figure 4.7, the condition yields sensible n. Further analysis is given right below the theorem.

Theorem 4.2.2
Let $R_+ \coloneqq R_1 + R_2$ be the sum of two independent random variables, $R_1, R_2 \sim \mathrm{Bin}(n, p_\mathrm{B})$. With the restrictions from Lemma 4.2.1, i.e. $\tau > np_\mathrm{B}$, $\hat{\tau} \coloneqq \frac{\tau}{n}$, $c \coloneqq \min\{\mathrm{e}^{-1}, \mathrm{D_{KL}}(\hat{\tau}||p_\mathrm{B})\}$ and $n > \frac{1}{2c}\mathrm{W}_{-1}(-c)$, it is

$$\Pr\{R_+ \geq 2\tau\} < \Pr\{R_2 \geq \tau\}.$$

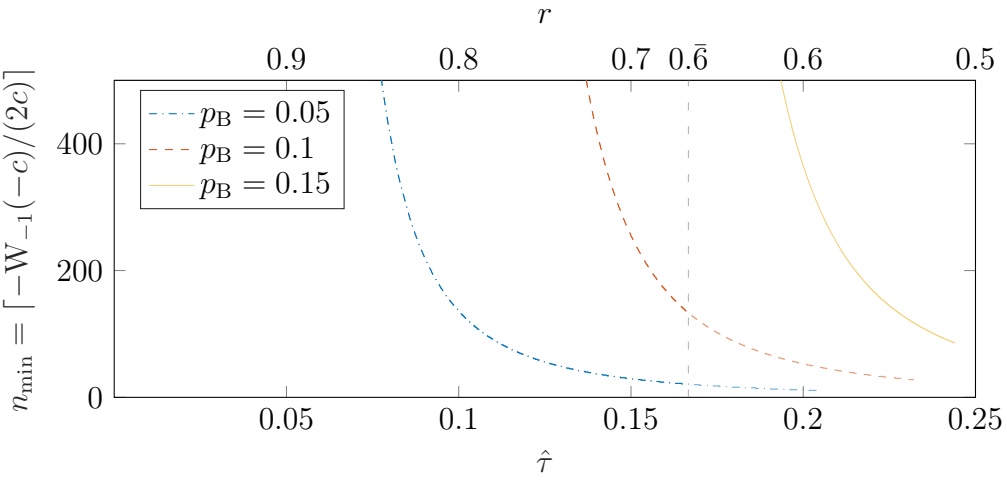


Figure 4.7.: Depiction of the lower bound on n for several values of p_B. The constant is chosen to be $c := \min\{e^{-1}, D_{KL}(\hat{\tau}||p_B)\}$. $r_{min,2} = 0.\bar{6}$, i.e. $R_D \geq R_L$ is only partly regarded, as indicated by the transparency. $W_{-1}(\cdot)$ is the -1st branch of the Lambert-W function.

Proof:
Regard the two bounds $(<)$ and $(>)$ from page 37. Since R_+ is distributed as $\text{Bin}(2n, p_B)$, cf. (4.1), we get

$$\begin{aligned}
\Pr\{R_+ \geq 2\tau\} &\overset{(<)}{\leq} \exp\left\{-2nD_{KL}\left(\frac{2\tau}{2n}||p_B\right)\right\} \\
&= \exp\left\{-nD_{KL}\left(\frac{\tau}{n}||p_B\right)\right\}^2 \\
&= \underbrace{\exp\left\{-nD_{KL}\left(\frac{\tau}{n}||p_B\right)\right\}}_{\overset{\text{Lemma 4.2.1}}{<} \frac{1}{\sqrt{2n}}} \cdot \exp\left\{-nD_{KL}\left(\frac{\tau}{n}||p_B\right)\right\} \\
&< \frac{1}{\sqrt{2n}} \exp\left\{-nD_{KL}\left(\frac{\tau}{n}||p_B\right)\right\} \\
&\overset{(>)}{\leq} \Pr\{R_2 \geq \tau\}.
\end{aligned}$$

□

The lower bound on n, we call it n_{min}, is illustrated in Figure 4.7. One can see that, for a given p_B, one can find an n such that the inequality is solved by increasing τ and n by the same factor, i.e. fix $\hat{\tau}$ and increase n. Note that $\hat{\tau}$ always refers to the correction radius of the lifted Gabidulin code. The closer τ gets to a considered p_B, the larger n must be chosen if one wants to guarantee a performance improvement of DLNC against lifting. For $\hat{\tau} \gg p_B \Rightarrow D_{KL}(\hat{\tau}||p_B) \uparrow$. Since $W_{-1}(-z) \notin \mathbb{R}$ for $z > e^{-1}$, the constant c is chosen to be $c := \min\{e^{-1}, D_{KL}(\hat{\tau}||p_B)\}$, which means for

$\hat{\tau} \to 1$ (where $\mathrm{D}_{\mathrm{KL}}(\hat{\tau}||p_{\mathrm{B}}) > \mathrm{e}^{-1}$ at some point), $n_{\min}$ becomes constant over $\hat{\tau}$. This region is not reached, due to the fact, that the minimum distance of a code cannot exceed the code length. This causes a condition for the minimum distance used in the code of DLNC

$$\begin{aligned} d_{\mathrm{rk,D}} = 2\tau_{\mathrm{D}} + 1 \leq n \quad &\overset{\tau_{\mathrm{D}}=2\tau}{\Longrightarrow} \quad 2\hat{\tau} \leq \frac{1}{2} - \frac{1}{2n} \\ &\Longleftrightarrow \hat{\tau} \leq \frac{1}{4} - \frac{1}{4n} \end{aligned}$$

and therefore a constraint for the abscissa. Note that $n \geq \tau$ is necessary for the definition of the Kullback-Leibler divergence, nevertheless this is satisfied by the code design, because otherwise the minimum distance $d_{\mathrm{rk,L}} > n$ and hence contradicts its definition. Since $\delta = 2$, the minimum code rate for the restriction $R_{\mathrm{D}} \geq R_{\mathrm{L}}$ (cf. Lemma 3.2.2) is $r_{\min,2} = \frac{2}{3}$. The overall rate is accordingly $R_{\mathrm{L}} = \frac{1}{3}$, see also (3.24) on p. 32.

4.3. Additional Consideration of the Receive Rank Deficiency Matrix

The additive error $\boldsymbol{E}_i$, derived for the AMC in DLNC by Seidl [SCFH13], is calculated by

$$\boldsymbol{E}_i = \boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\boldsymbol{S}_i - \boldsymbol{Y}_{i-1}^{+}\boldsymbol{B}_{i-1}\boldsymbol{S}_i + \boldsymbol{Y}_{i-1}^{+}\boldsymbol{B}_i.$$

Due to calculation rules for the rank of a matrix, cf. equations (2.2) and (2.3) from page 4, the error rank $\mathrm{rk}\,(\boldsymbol{E}_i)$ has been approached by

$$\mathrm{rk}\,(\boldsymbol{E}_i) \leq \mathrm{rk}\,(\boldsymbol{B}_{i-1}) + \mathrm{rk}\,(\boldsymbol{B}_i)\,,$$

since $\boldsymbol{S}_i$ and $\boldsymbol{Y}_{i-1}^{+}$ have full rank (cf. [SCFH13]). Up to now $\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}$ had been neglected. Now we consider this part. Regard the upper bound of $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ as introduced in (3.5) on p. 24:

$$\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq n - \mathrm{rk}\,(\boldsymbol{A}_i) + \mathrm{rk}\,(\boldsymbol{B}_i)\,.$$

If we assume $\mathrm{rk}\,(\boldsymbol{A}_i) = n$, we can upper bound $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq \mathrm{rk}\,(\boldsymbol{B}_i)$. The consideration of $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ therefore leads to a supplementary addend

$$\mathrm{rk}\,(\boldsymbol{E}_i) \leq \mathrm{rk}\,(\boldsymbol{B}_i) + \mathrm{rk}\,(\boldsymbol{B}_{i-1}) + \mathrm{rk}\,(\boldsymbol{B}_i) = \mathrm{rk}\,(\boldsymbol{B}_{i-1}) + 2\,\mathrm{rk}\,(\boldsymbol{B}_i)\,. \tag{4.5}$$

Consequently all components of the error rank of DLNC in the static case are considered and $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ is approximated, as seen in Figure 4.8.

From Theorem 4.2.2 we can deduce

$$\operatorname{rk}(\boldsymbol{E}_i) \leq \underbrace{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)}_{\leq \operatorname{rk}(\boldsymbol{B}_i)} + \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) + \underbrace{\operatorname{rk}(\Delta \boldsymbol{A}_i)}_{=0}$$

Figure 4.8.: Static case, where $\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq \operatorname{rk}(\boldsymbol{B}_i)$ is applied.

Corollary 4.3.1
Consider the following random variables $E \leq R_1 + 2R_2$, *where* $R_1, R_2 \sim \operatorname{Bin}(n, p_{\mathrm{B}})$. *Let* $\tau > np_{\mathrm{B}}$, $\hat{\tau} = \frac{\tau}{n}$, $c \coloneqq \min\{\mathrm{e}^{-1}, \mathrm{D}_{\mathrm{KL}}(\hat{\tau}||p_{\mathrm{B}})\}$ *and* $n > -\frac{1}{2c}\mathrm{W}_{-1}(-c)$. *Then*

$$\Pr\{E \geq 4\tau\} < \Pr\{R_2 \geq \tau\}.$$

Proof:
Since $E \leq R_1 + 2R_2 \leq 2(R_1 + R_2) = 2R_+$ with $R_+ \sim \operatorname{Bin}(2n, p_{\mathrm{B}})$, we have

$$\Pr\{E \geq 4\tau\} \leq \Pr\{2R_+ \geq 4\tau\} = \Pr\{R_+ \geq 2\tau\}.$$

Then because of Lemma 4.2.1 and Theorem 4.2.2, we can state

$$\Pr\{E \geq 4\tau\} < \Pr\{R_2 \geq \tau\} \text{ if } n > -\frac{\mathrm{W}_{-1}(-c)}{2c},$$

for $0 < c \leq \mathrm{D}_{\mathrm{KL}}(\tau/n||p)$. □

Thus we have proved, that DLNC exceeds RLNC with lifting for a factor of $\delta = 4$ between the error correction radii, which implies a higher correction radius for an overall rate $R_{\mathrm{L}} \geq 3/7$ (according to equation (3.24) on page 32). This bound for R_{L} can be improved by reducing δ. First we introduce another lemma.

Lemma 4.3.2
Let $\alpha \in (0, 1]$, $x_2 > x_1 > np_{\mathrm{B}}$, *further* $c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{x_1}{n}\middle||p_{\mathrm{B}}\right)$ *and*

$$c_2 \coloneqq \min\left\{\mathrm{D}_{\mathrm{KL}}\left(\frac{x_2}{n}\middle||p_{\mathrm{B}}\right), c_1 + \frac{\mathrm{e}^{-1}}{\alpha^2}\right\}.$$

If $n > \mathrm{W}_{-1}(\alpha^2(c_1 - c_2))/(2(c_1 - c_2))$, *then*

$$\exp\left\{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{x_2}{n}\middle||p_{\mathrm{B}}\right)\right\} < \frac{\alpha}{\sqrt{2n}}\exp\left\{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{x_1}{n}\middle||p_{\mathrm{B}}\right)\right\}.$$

Proof:
Analogous to the proof of Lemma 4.2.1, we have

$$n > \frac{1}{2(c_1 - c_2)}\mathrm{W}_{-1}(\alpha^2(c_1 - c_2)) \quad \Longrightarrow \quad \mathrm{e}^{2n(c_2 - c_1)} > \frac{2n}{\alpha^2},$$

where $n \in \mathbb{R}$ due to the condition $\alpha^2(c_2 - c_1) \leq \mathrm{e}^{-1}$, further $n > 0$ since $\alpha^2 > 0$ and $c_2 > c_1$, which emanates from $x_2 > x_1$ and the monotonicity of the Kullback-Leibler divergence. By repositioning the terms analogous to the proof of Lemma 4.2.1 we obtain

$$\exp\left\{-n\left[\mathrm{D}_{\mathrm{KL}}\left(\frac{x_2}{n}\middle||p_{\mathrm{B}}\right) - \mathrm{D}_{\mathrm{KL}}\left(\frac{x_1}{n}\middle||p_{\mathrm{B}}\right)\right]\right\} \leq \mathrm{e}^{-n(c_2-c_1)} < \frac{\alpha}{\sqrt{2n}},$$

since $c_2 \leq \mathrm{D}_{\mathrm{KL}}\left(\frac{x_2}{n}\middle||p_{\mathrm{B}}\right)$, from which the claim emerges directly. □

Note that the smaller α, the larger n must be chosen. With this lemma, we can state the following theorem. Again it is shown, that for large enough n and suitable ε_δ (especially not too small) the failure probability of DLNC is smaller than the one of the lifted Gabidulin code. The new condition can be observed in Figure 4.9. Further analysis is given below the theorem.

Theorem 4.3.3
Let $R_1, R_2 \sim \mathrm{Bin}\,(n, p_{\mathrm{B}})$. *Further* $\delta > 3$, $0 < \varepsilon_\delta \leq (\delta - 3)\tau/3$, $c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n}\middle||p_{\mathrm{B}}\right)$ *and* $c_2 \coloneqq \min\{\mathrm{D}_{\mathrm{KL}}\left(\frac{\tau+\varepsilon_\delta}{n}\middle||p_{\mathrm{B}}\right), c_1 + 4\mathrm{e}^{-1}\}$.[1] *If* $n > \frac{\mathrm{W}_{-1}((c_1-c_2)/4)}{2(c_1-c_2)}$, *then*

$$\Pr\{R_1 + 2R_2 \geq \delta\tau\} < \Pr\{R_2 \geq \tau\}.$$

Proof:
The proof is in Appendix A.1, p. 83. The idea of the proof is, to utilize the arguments of Theorem 4.2.2 on $2\Pr\{R_2 \geq \tau + \varepsilon_\delta\}$ instead of $\Pr\{R_+ \geq \tau\}$, which we get by splitting the probability $\Pr\{R_1 + 2R_2 \geq \delta\tau\}$, using the law of total probability. □

Considering $\mathrm{rk}\,(\boldsymbol{B}_i) = R_2$ and $\mathrm{rk}\,(\boldsymbol{B}_{i-1}) = R_1$, this theorem states that DLNC is better than lifting if ε_δ is chosen reasonably, i.e. not too small, and n large enough, where the lower bound on n is given by the Lambert W-function as in the conditions. This time, there is again the upper bound on the argument of the Lambert-W function, implying $c_2 - c_1 < 4\mathrm{e}^{-1}$. Similar to before, we choose

$$c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n}\middle||p_{\mathrm{B}}\right) \quad \wedge \quad c_2 \coloneqq c_1 + \frac{4}{\mathrm{e}},$$

for the case $\mathrm{D}_{\mathrm{KL}}\left(\frac{\tau+\varepsilon_\delta}{n}\middle||p_{\mathrm{B}}\right) > \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n}\middle||p_{\mathrm{B}}\right) + \frac{4}{\mathrm{e}}$, which occurs for $\varepsilon_\delta \to \tau$.

In Figure 4.9 one can see the lower bound on n for a certain p_{B}. It also shows that for a higher code rate r (especially $\hat{\tau} \to p_{\mathrm{B}}$) or lower choices of δ, n has to be chosen larger. Be aware, that $\delta\hat{\tau} \leq \frac{1}{2} - \frac{1}{2n}$, since the two procedures cannot be compared if the minimum distance of DLNC $d_{\mathrm{rk,D}} = 2\delta\tau + 1$ exceeds the code length n. Therefore one can see different dropouts before $\hat{\tau} = 0.15$. Furthermore

[1] The last definition guarantees $c_2 - c_1 \leq 4\mathrm{e}^{-1}$.

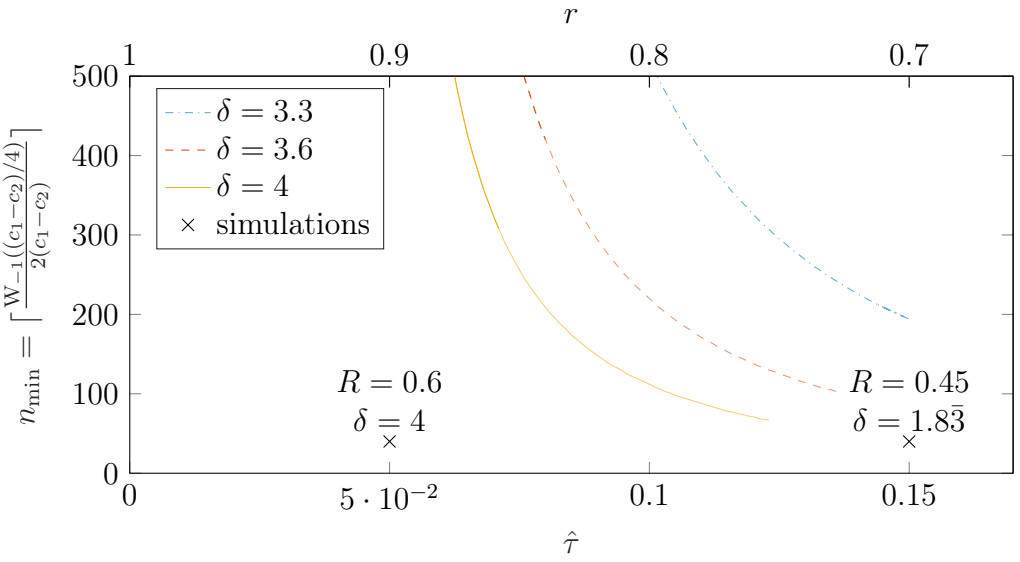


Figure 4.9.: Lower bound on n for several factors δ. Here $\hat{\tau} = \frac{\tau}{n}$, $p_{\mathrm{B}} = 0.05$, $c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n} \middle|\middle| p_{\mathrm{B}}\right)$ and $c_2 \coloneqq \min\left\{c_1 + \frac{4}{\mathrm{e}}, \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau+\varepsilon_\delta}{n} \middle|\middle| p_{\mathrm{B}}\right)\right\}$, where $\varepsilon_\delta = \left(\frac{\delta}{3} - 1\right)\tau$. The simulations (indicated by the marks) have been executed for $n = 40$. $R_{\mathrm{D}} \geq R_{\mathrm{L}}$ is only assumed for non-transparent parts (for $\delta = 3.3$ the restriction is not fulfilled at all).

$\tau + \varepsilon_\delta \leq n$, otherwise the argument of the Kullback-Leibler divergence does not fulfill its constraints. Nevertheless this case only occurs only for high ε_δ, respectively δ and therefore does not restrict the regions of our interest, see condition $\varepsilon_\delta \leq (\delta - 3)\frac{\tau}{3}$ from the theorem, which also yields

$$\delta \geq 3\left(1 + \frac{\varepsilon_\delta}{\tau}\right), \tag{4.6}$$

i.e. we achieve a smaller $\delta < 4 \iff \varepsilon_\delta < \tau/3$. The condition also implies $\delta \overset{!}{>} 3$, otherwise $\varepsilon_\delta \leq 0$, which is not allowed.

According to Lemma 3.2.2 (see p. 30), we can use $r_{\min,f}$ from equation (3.22) as bound in the plot, when restricting to $R_{\mathrm{D}} \geq R_{\mathrm{L}}$.[2] The other direction, i.e. getting δ from a minimum code rate, is done by

$$\begin{aligned} & & r_{\min,f} &= \frac{\delta_{\min} - 1}{\delta_{\min} - \frac{1}{f}} \\ \iff & & \delta_{\min} &= \frac{1 - \frac{r_{\min,f}}{f}}{1 - r_{\min,f}}, \end{aligned} \tag{4.7}$$

[2]This is only partly considered in Figure 4.9, i.e. $r < r_{\min,2}$, where the plots are transparent.

where f is defined as in (3.21) on page 31. By the claim of the theorem, the lower bound of the overall rate from (3.24) asymptotically reduces to

$$R_{\mathrm{L}} \geq \frac{\delta - 1}{2\delta - 1} \overset{(4.6)}{\geq} \frac{2 + 3\frac{\varepsilon_\delta}{\tau}}{5 + 6\frac{\varepsilon_\delta}{\tau}} \xrightarrow{\varepsilon_\delta \to 0} \frac{2}{5}.$$

Note that the bound for $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ is rather rough, therefore the lower bound on n is not tight. In the following result we can tighten the bound by keeping $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ as it is.

4.4. Lifting vs. DLNC in Static Networks

Finally all terms (cf. Figure 4.10) are regarded in the error rank of the DLNC procedure. From Section 2.3.3 we know that in the lifting case it must be

$$\mathrm{rk}\left(\boldsymbol{E}_i\right) \leq \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) + \mathrm{rk}\left(\boldsymbol{B}_{i-1}\right) + \mathrm{rk}\left(\boldsymbol{B}_i\right) + \underbrace{\mathrm{rk}\left(\Delta\boldsymbol{A}_i\right)}_{=0}$$

Figure 4.10.: All components of the error rank in the static case are considered.

$$d_{\mathrm{s}}(\langle \boldsymbol{X}_i \rangle, \langle \boldsymbol{Y}_i \rangle) = 2\,\mathrm{rk}\left(\boldsymbol{B}_i\right) + n - \mathrm{rk}\left(\boldsymbol{Y}_i\right) \leq d_{\mathrm{rk,L}} - 1$$

for decodability (cf. Theorem 2.3.5). Here we indexed the matrices with the respective generation in order to be able to compare to DLNC later. On the other hand for DLNC we have due to (3.4):

$$\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) = n - \mathrm{rk}\left(\boldsymbol{Y}_i\right)$$

and therefore the following condition must hold

$$2\,\mathrm{rk}\left(\boldsymbol{E}_i\right) \leq 2\left(\mathrm{rk}\left(\boldsymbol{B}_{i-1}\right) + \mathrm{rk}\left(\boldsymbol{B}_i\right) + n - \mathrm{rk}\left(\boldsymbol{Y}_i\right)\right) \leq d_{\mathrm{rk,D}} - 1.$$

One can see, that the RRDM $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) = n - \mathrm{rk}\left(\boldsymbol{Y}_i\right)$ appears in both formulas. We introduce ρ, with $d_{\mathrm{rk,D}} = \rho d_{\mathrm{rk,L}}$. It shall be the equivalent to δ from before, i.e.

$$\begin{aligned} & 2\tau_{\mathrm{D}} + 1 = \rho(2\tau + 1) \\ \Longleftrightarrow \qquad & \tau_{\mathrm{D}} = \rho\tau + \frac{\rho - 1}{2} = \left(\rho + \frac{\rho - 1}{2\tau}\right)\tau = \delta\tau. \end{aligned} \tag{4.8}$$

Therefore ρ can be calculated from δ and τ by

$$\rho = \frac{2\delta\tau + 1}{2\tau + 1} \overset{\tau\uparrow}{\approx} \delta.$$

As before we define $R_+ \coloneqq \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) = R_1 + R_2$. Now we want to prove:

$$\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2} d_{\mathrm{rk,L}}\right\} \leq \Pr\left\{2\operatorname{rk}(\boldsymbol{B}_i) + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\right\}. \tag{4.9}$$

The probabilities represent the failure probabilities of the two procedures. With the next theorem we find a tighter lower bound on n than in Theorem 4.3.3, for which the DLNC scheme supersedes the lifted Gabidulin scheme. Analysis and illustration of the bound can be found below the theorem and in Figure 4.11.

Theorem 4.4.1
Let $R_+ = R_1 + R_2$ be the sum of two binomially distributed random variables, i.e. $R_1, R_2 \sim \operatorname{Bin}(n, p_{\mathrm{B}})$, where $np_{\mathrm{B}} < \frac{d_{\mathrm{rk,L}}}{2} < \left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1$. Let $\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ be another random variable and $\alpha = \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\}$. Further $c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{d_{\mathrm{rk,L}}}{2n} \middle|\middle| p_{\mathrm{B}}\right)$, $c_2 \coloneqq \min\left\{\mathrm{D}_{\mathrm{KL}}\left(\frac{1}{n}\left(\left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1\right) \middle|\middle| p_{\mathrm{B}}\right), c_1 + \frac{\mathrm{e}^{-1}}{\alpha^2}\right\}$ and $n > \frac{\mathrm{W}_{-1}(\alpha^2(c_1 - c_2))}{2(c_1 - c_2)}$. Then

$$\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2} d_{\mathrm{rk,L}}\right\} < \Pr\left\{2R_2 + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\right\}.$$

Proof:
The proof is given in Appendix A.2 (on p. 85). It resembles the proof of Theorem 4.3.3 with the difference, that the law of total probability is applied on each side of the inequality, leading to the same addend on both sides, which can therefore be canceled. The rest is then application of Lemma 4.3.2. □

For the plots in Figure 4.11 we use the abbreviation $\hat{d} = d_{\mathrm{rk,L}}/n$ as equivalent for $\hat{\tau}$. Regard that there is a connection between $\hat{d}$ and code rate r:

$$\hat{d} = \frac{d_{\mathrm{rk,L}}}{n} \overset{\text{MRD code}}{=\!=\!=} \frac{n - k_{\mathrm{L}} + 1}{n} = 1 - r + \frac{1}{n},$$

and the connection depends on n. Looking at it from the other side we find that r is a function of $\hat{d}$ and n. We define

$$r(n) = 1 - \hat{d} + \frac{1}{n} \xrightarrow{n\to\infty} 1 - \hat{d}. \tag{4.10}$$

The approximation for $n \to \infty$, which is a lower bound on $r(n)$, is used in the plots. Furthermore, we approximate $\hat{x}_1 = \left(\frac{\rho}{2} - 1\right)\hat{d} + \frac{1}{n}$ by $\tilde{x}_1 = \left(\frac{\rho}{2} - 1\right)\hat{d}$, which is a lower bound, for the plots. So the actual $n_{\min}$ is slightly smaller. It yields a familiar

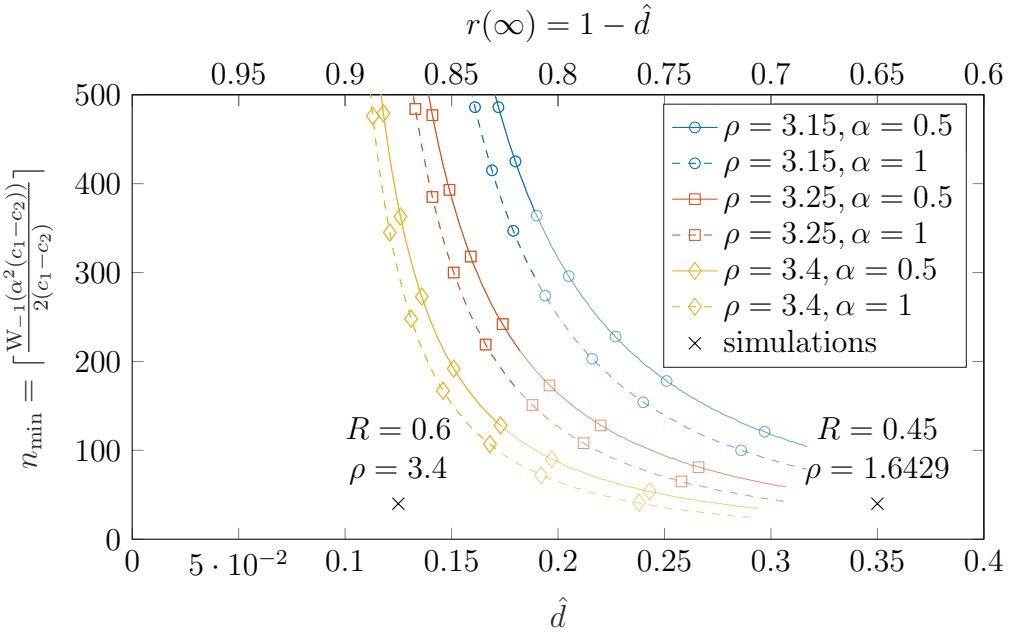


Figure 4.11.: Lower Bound for n according to Theorem 4.4.1 for several ρ and α with parameters $p_{\mathrm{B}} = 0.05$, $c_1 = \mathrm{D}_{\mathrm{KL}}\left(\frac{1}{2}\hat{d}\middle||p_{\mathrm{B}}\right)$, $c_2 = \min\left\{c_1 + \frac{1}{\alpha^2 \mathrm{e}}, \mathrm{D}_{\mathrm{KL}}\left(\tilde{x}_1\middle||p_{\mathrm{B}}\right)\right\}$, where $\tilde{x}_1 = \left(\frac{\rho}{2} - 1\right)\hat{d} \leq \left(\frac{\rho}{2} - 1\right)\hat{d} + \frac{1}{n}$. $r(\infty)$ in the upper x-axis is a lower bound for the code rate that can be achieved at a specific $\hat{d}$, cf. (4.10). Note that $R_{\mathrm{D}} \geq R_{\mathrm{L}}$ is not regarded in the transparent part of the curves.

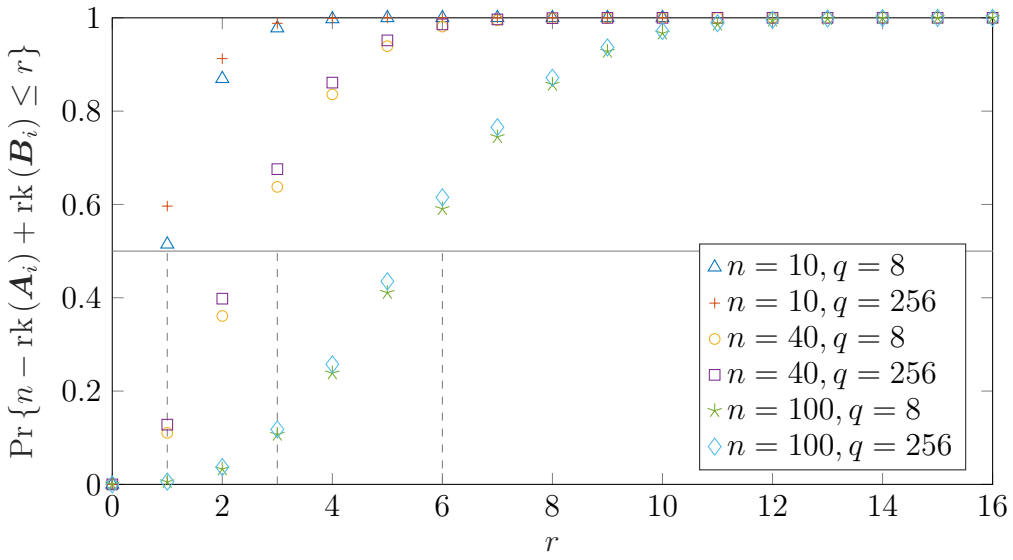


Figure 4.12.: 0.5-Quantiles of the lower bound of the CDF of $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ as given in (3.6) on p. 25 for $q = 8$. The respective relative ranks are $r_1 = \frac{r}{n} = \frac{1}{10} = 0.1$, $r_2 = \frac{3}{40} = 0.075$, $r_3 = \frac{6}{100} = 0.06$. Here $p_{\mathrm{B}} = 0.05$.

behavior, the smaller α and the smaller ρ, the larger one has to choose n. As usual, when $\hat{\tau}$ approximates p_{B}, n must be larger too. Here $\rho \cdot \hat{d} \leq 1$ must hold, for lifting and DLNC to be comparable. The depicted simulations are the same as before, but, according to our choice of k_{L} and k_{D}, we calculated ρ instead of δ. Figure 3.1 on page 25 shows that for $\hat{d} > 0.1$ the assumption $\alpha = \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\} \geq 0.5$ is a suitable assumption. The same figure is shown in more detail as Figure 4.12, where the approximate median, i.e. 0.5-quantile of some of the CDFs is also drawn.

Due to the codomain of the Lambert-W function $\alpha^2(c_2 - c_1) \leq \mathrm{e}^{-1}$ must hold, thus c_2 is chosen to be

$$c_2 = \min\left\{c_1 + \frac{1}{\alpha^2 \mathrm{e}}, \mathrm{D}_{\mathrm{KL}}\left(\frac{1}{n}\left(\left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1\right) \middle|\middle| p_{\mathrm{B}}\right)\right\}.$$

One should note, that from the restrictions in Theorem 4.4.1, we get a constraint for ρ:

$$\begin{aligned} & \frac{d_{\mathrm{rk,L}}}{2} < \left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1 \\ \iff \qquad & \rho > 3 - \frac{2}{d_{\mathrm{rk,L}}}, \end{aligned} \tag{4.11}$$

i.e. the smaller $d_{\mathrm{rk,L}}$, the smaller one can choose ρ. For high values of $d_{\mathrm{rk,L}}$, ρ can be chosen close to and equal to 3, $\rho = 3$ yields $\delta = 3 + \frac{1}{\tau}$. By defining f as in

equation (3.21), see Section 3.2 p. 31, for $R_{\mathrm{D}} \geq R_{\mathrm{L}}$, the minimal ρ necessary can be calculated from code rate r by

$$r_{\min,f} \overset{(3.22)}{=} \frac{\delta - 1}{\delta - \frac{1}{f}} \overset{(4.8)}{=} \frac{\rho_{\min} + \frac{\rho_{\min}-1}{2\tau} - 1}{\rho_{\min} + \frac{\rho_{\min}-1}{2\tau} - \frac{1}{f}} =: r_{\rho} \tag{4.12}$$

$$\iff \qquad \rho_{\min} = \frac{1 - \frac{\frac{1}{2\tau}+\frac{1}{f}}{\frac{1}{2\tau}+1} r_{\min,f}}{1 - r_{\min,f}}. \tag{4.13}$$

Be aware, that due to (4.11), the code rate must fulfill

$$r_{\min,f} > \frac{2(1 - \frac{1}{2\tau+1})}{3 - \frac{\frac{1}{2\tau}+\frac{1}{f}}{\frac{1}{2\tau}+1} - \frac{2}{2\tau+1}}$$

for a specified τ. The respective overall rate with constraints $m = n$ and $q \to \infty$ is

$$\begin{aligned} R_{\mathrm{L}} &\overset{(3.24)}{=} \frac{\delta - 1}{2\delta - 1} = \frac{1}{2} \frac{\rho_{\min} + \frac{\rho_{\min}-1}{2\tau} - 1}{\rho_{\min} + \frac{\rho_{\min}-1}{2\tau} - \frac{1}{f}} \\ &\overset{\rho=3}{=} \frac{2 + \frac{1}{\tau}}{5 + \frac{2}{\tau}} \overset{\tau\uparrow}{\to} \frac{2}{5}, \end{aligned}$$

which is the same rate bound as in the result before. Note that, for high code rates and small n, the correction radius τ cannot be neglected. It is also important to note, that Figure 4.11 does not fully comply with the condition $R_{\mathrm{D}} \geq R_{\mathrm{L}}$, i.e. r contradicts $r_{\min,2}$ in the transparent areas.

However, one can see that DLNC exceeds lifting in this context again, when the parameters are chosen appropriately.

5. Slowly Varying Networks

Finally, we come to the part that justifies the term "slowly-varying" in the title, i.e. we regard the time-varying case. While [SCFH13] stuck to an invariant network, [PCF+15] allowed some changes in the network between generations. The changing network can be represented by the so-called *channel deviation* $\Delta \boldsymbol{A}_i$ (cf. Definition 2.4.1). In order to investigate time-varying networks, we therefore have to examine $\mathrm{rk}\,(\Delta \boldsymbol{A}_i)$ and allow especially $\mathrm{rk}\,(\Delta \boldsymbol{A}_i) > 0$.

5.1. Examination of the Channel Deviation

In Section 2.4.1 we have already described some properties of the channel deviation, as e.g. an upper bound and arising from that bound its PMF. One can also calculate the Probability Generating Function (PGF) of the upper bound of random variable $\mathrm{rk}\,(\Delta \boldsymbol{A}_i)$, which we denote by $\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)}$.

Lemma 5.1.1
Let $L \sim \mathrm{Bin}\,(|\mathcal{N}|, p_{\Delta\mathcal{N}})$, $W_j \overset{\text{i.i.d.}}{\sim} \mathrm{Bin}\,(|\mathcal{N}|, p_w)$ *and* $\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)} = \sum_{j=1}^{L} W_j$ *be random variables,* $|\mathcal{N}|, p_w$ *and* $p_{\Delta\mathcal{N}}$ *as defined in Section 2.4.1, cf. Table 2.3. Then the PGF of* $\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)}$*, we call it* $G_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}}$*, is given by*

$$G_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}}(z) = \left(1 - p_{\Delta\mathcal{N}} + p_{\Delta\mathcal{N}}(1 - p_w + p_w z)^{|\mathcal{N}|}\right)^{|\mathcal{N}|}. \tag{5.1}$$

Proof:
For a discrete random variable X the PGF is defined to be

$$G_X(z) \coloneqq \mathbb{E}[z^X] = \sum_{x=0}^{\infty} f_X(x) z^x. \tag{5.2}$$

For definition and properties of the PGF see [Spo14, Section 5.2]. One important property of the PGF of the sum of L independent identically distributed random variables $X_i \sim X$ is

$$G_{\sum_{i=1}^{L} X_i}(z) = (G_X(z))^L. \tag{5.3}$$

With these formulas we can derive

$$\begin{aligned} G_{\sum_{i=0}^{L} W}(z) &\overset{(5.2)}{=\!=} \mathbb{E}\left[z^{\sum_{i=0}^{L} W}\right] \\ &\overset{\text{law of total expectation}}{=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=} \mathbb{E}\left[\mathbb{E}\left[z^{\sum_{i=0}^{L} W} | L\right]\right] \\ &\overset{(5.3)}{=\!=} \mathbb{E}\left[(G_W(z))^L\right] \\ &\overset{(5.2)}{=\!=} G_L(G_W(z)). \end{aligned}$$

Since the PGF of a binomial random variable $X \sim \mathrm{Bin}\,(n,p)$ reads

$$G_X(z) = (1-p+pz)^n,$$

it is

$$G_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}}(z) = \left(1 - p_{\Delta\mathcal{N}} + p_{\Delta\mathcal{N}}(1-p_w+p_w z)^{|\mathcal{N}|}\right)^{|\mathcal{N}|}.$$

□

The knowledge about the PGF is useful for the calculation of i-th moment of a random variable, since via the Moment Generating Function (MGF) $M_X(t)$ it is

$$\mathbb{E}\left[X^i\right] = \frac{\mathrm{d}^i}{\mathrm{d}t^i} M_X(t)\Big|_{t=0} = \frac{\mathrm{d}^i}{\mathrm{d}t^i} G_X(\mathrm{e}^t)\Big|_{t=0}.$$

Therefore we have as expected value of $\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)}$

$$\begin{aligned} \mathbb{E}_{\boldsymbol{A}} \coloneqq \mathbb{E}\left[\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)}\right] &= \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - p_{\Delta\mathcal{N}} + p_{\Delta\mathcal{N}}\left(1-p_w+p_w\mathrm{e}^t\right)^{|\mathcal{N}|}\right)^{|\mathcal{N}|}\Big|_{t=0} \\ &= |\mathcal{N}|\left(1 - p_{\Delta\mathcal{N}} + p_{\Delta\mathcal{N}}\left(1-p_w+p_w\mathrm{e}^t\right)^{|\mathcal{N}|}\right)^{|\mathcal{N}|-1} \cdot \\ &\quad \cdot p_{\Delta\mathcal{N}} \cdot |\mathcal{N}|\left(1-p_w+p_w\mathrm{e}^t\right)^{|\mathcal{N}|-1} \cdot p_w \mathrm{e}^t\Big|_{t=0} \\ &= |\mathcal{N}|^2 p_{\Delta\mathcal{N}} p_w, \end{aligned}$$

which is the same as the multiplication of the expected value of random variable L and the one of random variable W. Using the Markov bound, we obtain

$$\Pr\left\{\mathrm{rk}\,(\Delta \boldsymbol{A}_i) > \varepsilon_A\right\} \leq \Pr\left\{\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)} > \varepsilon_A\right\} \leq \frac{\mathbb{E}_{\boldsymbol{A}}}{\varepsilon_A} = \frac{|\mathcal{N}|^2 p_{\Delta\mathcal{N}} p_w}{\varepsilon_A}. \tag{5.4}$$

With higher moments and for example the Chernoff bound, it could be possible to derive tighter bounds. In Section 2.4.1, more precisely equation (2.19) on p. 19, we have defined the PMF of an upper bound of $\mathrm{rk}\,(\Delta \boldsymbol{A}_i)$. The CDF denoted by $F_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}}(x)$ can be achieved by adding up the values of the PMF. Quantiles of

the random variable are calculated via the quantile function $F^{-1}_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}} : [0,1] \to \mathbb{R}$, defined as $F^{-1}_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}}(y) = \inf\{x : F_{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}}(x) \geq y\}$. Consequently

$$\begin{aligned} & \Pr\left\{\mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) > \varepsilon_A\right\} \leq \Pr\left\{\overline{\mathrm{rk}\left(\Delta \boldsymbol{A}_i\right)} > \varepsilon_A\right\} \leq c \\ \iff & \qquad\qquad \varepsilon_A \geq F^{-1}_{\mathrm{rk}(\Delta \boldsymbol{A}_i)}(1-c). \end{aligned}$$

This will be used later on.

5.2. Probabilistic Analysis

As in Chapter 4, we want to compare the failure probabilities of DLNC and lifting. Additional to the chapter before we consider the channel deviation, which compromises the performance of DLNC. Section 5.2.1 regards the problem without any approximations for the error ranks, while in Section 5.2.2 we use the theorems introduced in Chapter 4 and integrate them in the new setting.

5.2.1. General View

As DLNC is not suited for heavily changing networks, it is useful to know the boundary between networks that change too much for DLNC to be applied and the ones which change slowly enough to have a benefit with DLNC compared to lifting. In order to find this boundary, we introduce the following definition.

Definition 5.2.1 (ε-varying Network)
Let $\mathrm{rk}\left(\boldsymbol{E}_i\right)$ *refer to the error rank when using DLNC and a Gabidulin code with error correction radius* $\delta\tau$ *in a static network. The network is then called* ε*-varying if* $\exists\, \varepsilon > 0$:

$$\Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) + \mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) \geq \delta\tau\right\} \leq \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) \geq \delta\tau\right\} + \varepsilon.$$

From the definition we directly get the following lemma, which gives a rough approximation.

Lemma 5.2.2
A network with channel deviation $\Delta \boldsymbol{A}_i$ *is* ε*-varying if*

$$\Pr\left\{\mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) > 0\right\} \leq \varepsilon.$$

Proof:

$$\begin{aligned}
\Pr\{\mathrm{rk}(\boldsymbol{E}_i) + \mathrm{rk}(\Delta\boldsymbol{A}_i) \geq \delta\tau\} &= \sum_{i=0}^{n} \Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) = i\} \Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau - i\} \\
&\leq \underbrace{\Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) = 0\}}_{\leq 1} \Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau\} + \\
&\quad + \sum_{i>0} \Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) = i\} \underbrace{\Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau - i\}}_{\leq 1} \\
&\leq \Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau\} + \Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) > 0\} \\
&\leq \Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau\} + \varepsilon. \qquad \square
\end{aligned}$$

This lemma highlights the importance of the probability $\Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) > 0\}$. Note that since $\Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) > 0\} \leq \Pr\left\{\overline{\mathrm{rk}(\Delta\boldsymbol{A}_i)} > 0\right\}$, it is

$$\Pr\left\{\overline{\mathrm{rk}(\Delta\boldsymbol{A}_i)} > 0\right\} \leq \varepsilon \quad \Longrightarrow \quad \Pr\{\mathrm{rk}(\Delta\boldsymbol{A}_i) > 0\} \leq \varepsilon.$$

Due to the property of the PGF, that $\Pr\{X = k\} = \frac{1}{k!} G_X^{(k)}(0)$ for $k \in \mathbb{N}$, it is

$$\begin{aligned}
\Pr\left\{\overline{\mathrm{rk}(\Delta\boldsymbol{A}_i)} > 0\right\} &= 1 - \Pr\left\{\overline{\mathrm{rk}(\Delta\boldsymbol{A}_i)} = 0\right\} = 1 - G_{\overline{\mathrm{rk}(\Delta\boldsymbol{A}_i)}}(0) \\
&= 1 - \left(1 - p_{\Delta\mathcal{N}} + p_{\Delta\mathcal{N}}(1 - p_w)^{|\mathcal{N}|}\right)^{|\mathcal{N}|}.
\end{aligned}$$

Figure 5.1 shows the dependence of $\Pr\left\{\overline{\mathrm{rk}(\Delta\boldsymbol{A}_i)} > 0\right\}$ on the number of nodes $|\mathcal{N}|$ and the probability of a node to join or leave $p_{\Delta\mathcal{N}}$.

Let $\boldsymbol{E}_{\mathrm{LiftedGab}}$ be the error matrix in an RLNC channel, where lifting has been applied together with a Gabidulin code. In Chapter 4 it has been shown, that

$$\Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau\} < \Pr\{\mathrm{rk}(\boldsymbol{E}_{\mathrm{LiftedGab}}) \geq \tau\}$$

for several manifestations of $\mathrm{rk}(\boldsymbol{E}_i)$ and $\mathrm{rk}(\boldsymbol{E}_{\mathrm{LiftedGab}})$. Hence

$$\exists\varepsilon > 0 : \Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau\} + \varepsilon \leq \Pr\{\mathrm{rk}(\boldsymbol{E}_{\mathrm{LiftedGab}}) \geq \tau\}.$$

Thus, for an ε-varying network with $\varepsilon \leq \Pr\{\mathrm{rk}(\boldsymbol{E}_{\mathrm{LiftedGab}}) \geq \tau\} - \Pr\{\mathrm{rk}(\boldsymbol{E}_i) \geq \delta\tau\}$, DLNC improves upon lifting. It is hence suitable to define this as the boundary of slowly-varying networks.

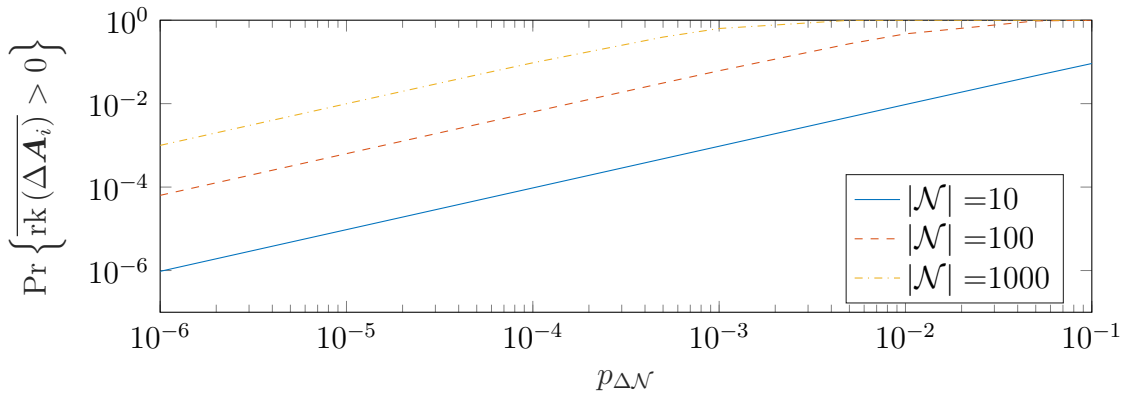


Figure 5.1.: Relation between $\Pr\left\{\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)} > 0\right\}$ and $p_{\Delta\mathcal{N}}$ for several network sizes $|\mathcal{N}|$. p_w was chosen to be 0.01.

Definition 5.2.3 (Slowly-Varying Network)
Let $\boldsymbol{E}_{\mathrm{LiftedGab}}$ and $\boldsymbol{E}_i$ be the error matrices in an ε-varying network, where a lifted Gabidulin code with error correction radius τ, respectively DLNC in combination with a Gabidulin code of error correction radius $\delta\tau$ have been applied. The network is called a slowly-varying network if

$$\varepsilon \leq \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_{\mathrm{LiftedGab}}\right) \geq \tau\right\} - \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) \geq \delta\tau\right\}.$$

From this definition and Lemma 5.2.2 we can build the next theorem.

Theorem 5.2.4
Regard a $\Pr\{\mathrm{rk}(\Delta \boldsymbol{A}_i) > 0\}$-varying network. Further let the network be slowly-varying, then

$$\Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) + \mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) \geq \delta\tau\right\} \leq \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_{\mathrm{LiftedGab}}\right) \geq \tau\right\}.$$

Proof:
By Definition 5.2.1 it is

$$\Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) + \mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) \geq \delta\tau\right\} \leq \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) \geq \delta\tau\right\} + \Pr\left\{\mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) > 0\right\}$$

and since the condition for slowly-varying is

$$\Pr\left\{\mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) > 0\right\} \leq \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_{\mathrm{LiftedGab}}\right) \geq \tau\right\} - \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) \geq \delta\tau\right\},$$

the proof directly follows, as

$$\begin{aligned}\Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) + \mathrm{rk}\left(\Delta \boldsymbol{A}_i\right) \geq \delta\tau\right\} &\leq \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) \geq \delta\tau\right\} + \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_{\mathrm{LiftedGab}}\right) \geq \tau\right\} \\ &\quad - \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_i\right) \geq \delta\tau\right\} \\ &= \Pr\left\{\mathrm{rk}\left(\boldsymbol{E}_{\mathrm{LiftedGab}}\right) \geq \tau\right\}.\end{aligned}$$

□

So far we treated the problem in a general way. When taking into account the components, that the error matrices consist of, the demand $\Pr\{\mathrm{rk}(\Delta \boldsymbol{A}_i) > 0\} \leq \varepsilon$ from Lemma 5.2.2 can be relaxed to $\Pr\{\mathrm{rk}(\Delta \boldsymbol{A}_i) > \varepsilon_A\} \leq \varepsilon$ for a $\varepsilon_A > 0$.

5.2.2. Extension of Theorem 4.3.3

We can specify the problem by using assumptions from the chapter before, e.g. $\mathrm{rk}(\boldsymbol{E}_i) \leq \mathrm{rk}(\boldsymbol{B}_{i-1}) + 2\,\mathrm{rk}(\boldsymbol{B}_i)$, cf. equation (4.5) on page 43. As before we define $R_1 = \mathrm{rk}(\boldsymbol{B}_{i-1})$ and $R_2 = \mathrm{rk}(\boldsymbol{B}_i)$, furthermore $\mathrm{rk}(\Delta \boldsymbol{A}_i)$ shall be named A. See Figure 5.2 for comparison to the actual error rank.

$$\mathrm{rk}(\boldsymbol{E}_i) \leq \underbrace{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)}_{\leq \mathrm{rk}(\boldsymbol{B}_i)} + \mathrm{rk}(\boldsymbol{B}_{i-1}) + \mathrm{rk}(\boldsymbol{B}_i) + \mathrm{rk}(\Delta \boldsymbol{A}_i)$$

Figure 5.2.: Varying case with the approximation of $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$.

In the following theorem we regard, which code rate region (specified by δ) has to be chosen, in order for the failure probability of DLNC to be smaller than the failure probability of lifting, when the network statistics and especially the statistics of $\mathrm{rk}(\Delta \boldsymbol{A}_i)$ is given. The discussion of parameters is given below the theorem. Later we consider the other way, i.e. which specifications the statistics of $\mathrm{rk}(\Delta \boldsymbol{A}_i)$ must fulfill for a given rate region. This interpretation is shown in Figure 5.4 and 5.5.

Theorem 5.2.5
Let $R_1, R_2 \sim \mathrm{Bin}(n, p_\mathrm{B})$, let A be a random variable with the PGF given in (5.1).[1] *Further let $np_\mathrm{B} < \tau < n$, $\delta' > 3$, $0 < \varepsilon_\delta \leq (\delta' - 3)\tau/3$, $c_1 \coloneqq \mathrm{D_{KL}}\left(\frac{\tau}{n}\middle||p_\mathrm{B}\right)$ and $c_2 \coloneqq \min\left\{\mathrm{D_{KL}}\left(\frac{\tau+\varepsilon_\delta}{n}\middle||p_\mathrm{B}\right), c_1 + 4\mathrm{e}^{-1}\right\}$. It must be $n > \frac{1}{2(c_1-c_2)}\mathrm{W}_{-1}\left(\frac{1}{4}(c_1 - c_2)\right)$ (as in Theorem 4.3.3). Further let $c \coloneqq \frac{1}{\sqrt{2n}}\mathrm{e}^{-nc_1} - 2\mathrm{e}^{-nc_2}$ and $\varepsilon_A = F_A^{-1}(1-c)$, where $F_A^{-1} : [0,1] \to \mathbb{R}$ is the quantile function of RV A, and $\delta = \delta' + \frac{\varepsilon_A}{\tau}$. Then*

$$\Pr\{R_1 + 2R_2 + A \geq \delta\tau\} \leq \Pr\{R_2 \geq \tau\}.$$

Proof:
The proof of this theorem is located in Appendix A.3, see p. 86. The proof works as the one for Theorem 4.3.3 to which it is the analog for varying networks. The

[1] i.e. specified by parameters $|\mathcal{N}|, p_w$ and $p_{\Delta\mathcal{N}}$

additional constraints guarantee, that the probability $\Pr\{A > \varepsilon_A\}$, which appears as new addend from the sum splitting after applying the law of total probability, is smaller than the difference of the failure probabilities assumed for Theorem 4.3.3, i.e. $\Pr\{R_2 \geq \tau\} - 2\Pr\{R_2 \geq \tau + \varepsilon_\delta\}$. □

Most of the conditions are familiar from Theorem 4.3.3. By $\varepsilon_A = F_A^{-1}(1-c)$, the new feature of varying networks is met.

The combined restrictions on ε_δ and δ are due to $\delta = \delta' + \frac{\varepsilon_A}{\tau}$ and $\varepsilon_\delta \leq (\delta' - 3)\frac{\tau}{3}$: $\varepsilon_\delta \leq (\delta - 3 - \varepsilon_A/\tau)\frac{\tau}{3}$ and

$$\delta \geq \frac{\varepsilon_A}{\tau} + 3\left(1 + \frac{\varepsilon_\delta}{\tau}\right) \tag{5.5}$$

respectively. From the choice of ε_δ and ε_A we can therefore compute a minimal value for δ, to find out the rate region (using Lemma 3.2.2), where DLNC exceeds lifting for the given network and the constraint $R_\mathrm{D} \geq R_\mathrm{L}$. This means, that ε_A and ε_δ are now sizing δ, but their task divides up. ε_A causes dimensioning of the probability of $\mathrm{rk}\,(\Delta \boldsymbol{A}_i)$ to be greater than ε_A, while ε_δ is necessary for the estimation of n. ε_A should be chosen large in order to minimize the probability $\Pr\{\mathrm{rk}\,(\Delta \boldsymbol{A}_i) > \varepsilon_A\}$, but ε_δ should as well be chosen large, if one wants to minimize n.

In the upcoming plots (cf. Figure 5.3 - 5.5) the following constraints must be fulfilled:

- $\hat{\tau} > p_\mathrm{B}$,
- $\frac{\tau+\varepsilon_\delta}{n} = \hat{\tau}(1+\tilde{\varepsilon}_\delta) \leq 1$ for it must be a p-coin in the Kullback-Leibler divergence,
- the argument of the Lambert-W function must be greater than $-\mathrm{e}^{-1}$ for real results,
- the minimum distance of DLNC must not exceed n, i.e. $\hat{\tau} \leq \frac{n-1}{2\delta n}$ and
- the code rate given by $\hat{\tau}$, namely $r = 1 - 2\hat{\tau}$ must be higher than $r_{\min,2} = \frac{\delta-1}{\delta-\frac{1}{2}}$, cf. (3.23), p. 32. Then the statements in the plot hold for $n = m$ and $R_\mathrm{D} \geq R_\mathrm{L}$.

The last point is the crucial one in the plots given below.

Remark
As it can be seen in inequality (A.15)*, see p. 88, in the proof of Theorem 5.2.5,* $c = \frac{1}{\sqrt{2n}}\exp\{-nc_1\} - 2\exp\{-nc_2\}$ *is a lower bound for the difference that specifies a slowly-varying network. It therefore plays a similar role to* ε *in slowly-varying networks. Due to* $\varepsilon_A > 0$*, the considered networks do not fulfill the* ε*-criterion, i.e.* $c \geq \varepsilon$*, but it could nevertheless be shown, that DLNC can outperform lifting in these networks.*

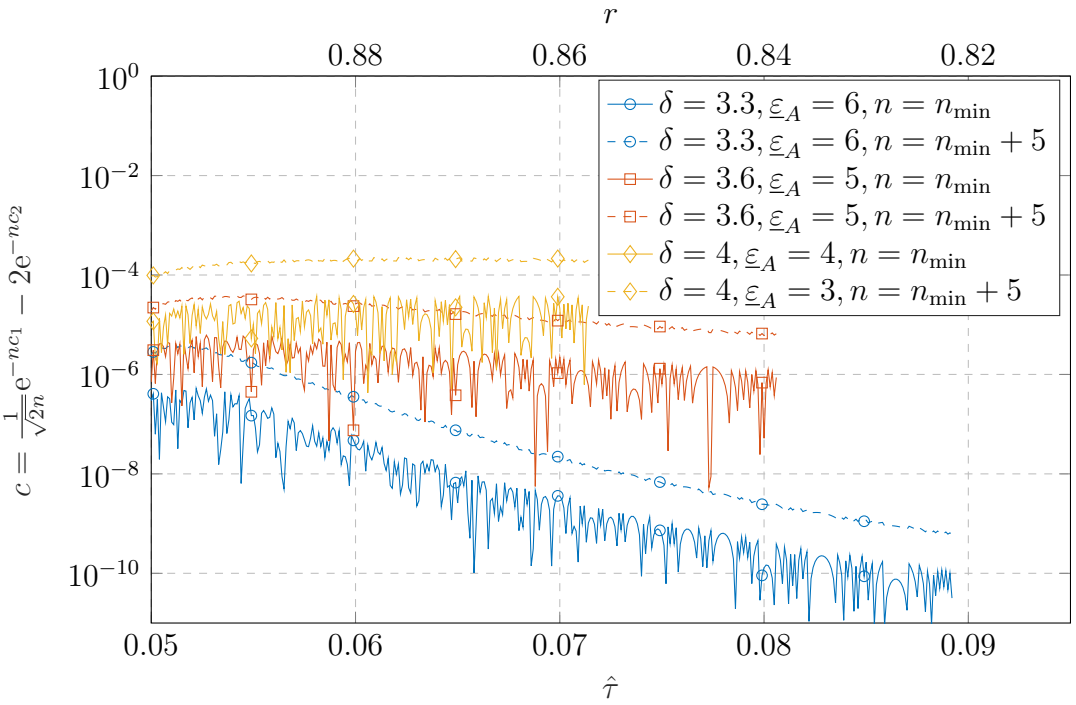


Figure 5.3.: Depiction of the difference c for several δ. Parameters: $p_B = 0.05$, $\tilde{\varepsilon}_A = \frac{\varepsilon_A}{\tau} = \frac{\delta-3}{4} = \tilde{\varepsilon}_\delta$, $n_{\min} = \lceil W_{-1}((c_1 - c_2)/4)/(2(c_1 - c_2)) \rceil$, where $c_1 = D_{KL}\left(\hat{\tau} \middle|\middle| p_B\right)$ and $c_2 = \min\left\{D_{KL}\left(\hat{\tau}(1+\tilde{\varepsilon}_\delta) \middle|\middle| p_B\right), c_1 + 4e^{-1}\right\}$. The step width between calculated values of $\hat{\tau}$ is 0.0001. Since ε_A (calculated as in (5.6)) is different for each point $\hat{\tau}$ in the plot, we only display its minimum (denoted by $\underline{\varepsilon}_A$) per curve.

Figure 5.3 shows, how the difference c might look like for several δ. Obviously, the closer δ is to 3, the smaller is c. One can see that the difference might change for varied choices of n. As parameters we chose $p_B = 0.05$, $\tilde{\varepsilon}_A = \frac{\varepsilon_A}{\tau} = \frac{\delta-3}{4}$ and $\tilde{\varepsilon}_\delta = 3\frac{\varepsilon_A}{\tau}$. Note that we chose δ, $\tilde{\varepsilon}_\delta = \frac{\varepsilon_\delta}{\tau}$ and $\tilde{\varepsilon}_A = \frac{\varepsilon}{\tau}$ with its fixed relation, so ε_A is calculated as

$$\varepsilon_A = \lfloor \tilde{\varepsilon}_A \cdot \lfloor \hat{\tau} \cdot n \rfloor \rfloor \tag{5.6}$$

at each point, therefore we only mention the minimum $\underline{\varepsilon}_A$. The pattern of jumps in the curves are due to the calculation of $n_{\min}$ for each point. Between the points we have steps of 0.0001. Since the curves are smoother for $n = n_{\min} + 5$, we use this in further plots.

It is also possible to look from another point of view on the problem by providing bounds for the parameters of the random variable $\mathrm{rk}(\Delta \boldsymbol{A}_i)$ for a certain correction radius τ and a rate region specified by δ. As it is a consequence of the theorem before, we commit it to writing as a corollary.

Corollary 5.2.6
Let $R_1, R_2 \sim \mathrm{Bin}(n, p_\mathrm{B})$ *be independent RVs,* $\tau > np_\mathrm{B}$, $\delta > 3$, $\varepsilon_\delta \in (0, \frac{\delta-3}{3}\tau)$ *and* $\varepsilon_A = (\delta-3)\tau - 3\varepsilon_\delta$. *Further* $c_1 = \mathrm{D_{KL}}\left(\frac{\tau}{n}\middle\|p_\mathrm{B}\right)$, $c_2 = \min\left\{\mathrm{D_{KL}}\left(\frac{\tau+\varepsilon_\delta}{n}\middle\|p_\mathrm{B}\right), c_1 + 4\mathrm{e}^{-1}\right\}$ *and* $n > \mathrm{W}_{-1}((c_1-c_2)/4)/(2(c_1-c_2))$ *(defined as in Theorem 4.3.3). In addition* $c = \frac{1}{\sqrt{2n}}\mathrm{e}^{-nc_1} - 2\mathrm{e}^{-nc_2}$. *If the parameters of RV A fulfill* $F_A^{-1}(1-c) \leq \varepsilon_A$, *then*

$$\Pr\{R_1 + 2R_2 + A \geq \delta\tau\} \leq \Pr\{R_2 \geq \tau\}.$$

Proof:
With $\delta > 3$, $\varepsilon_\delta \in (0, \frac{\delta-3}{3}\tau)$ and $\varepsilon_A = (\delta-3)\tau - 3\varepsilon_\delta$ we ensure that (5.5) holds with equality. Choosing n to be $n_\mathrm{min} = \lceil \mathrm{W}_{-1}((c_1-c_2)/4)/(2(c_1-c_2))\rceil$ (with the constants c_1 and c_2 as defined above) or higher and difference c, as given, together with $F_A^{-1}(1-c) \leq \varepsilon_A$, we have all the conditions from Theorem 5.2.5. □

The corollary describes, how the variations in the network are restricted, when we demand DLNC to overtop lifting for a certain rate region. Specifying the rate region by $r_{\mathrm{min},f}$ and using the parameter to calculate the respective δ_min is possible via equation (4.7) on p. 46.

The restriction on the changes in the network is given as an upper bound on the $(1-c)$-quantile, i.e. $F_A^{-1}(1-c) \leq \varepsilon_A$. Since the inversion of the CDF of $\overline{\mathrm{rk}(\Delta \boldsymbol{A}_i)}$ for any of the parameters $|\mathcal{N}|$, $p_{\Delta\mathcal{N}}$ or p_w is not trivial, one might want to use a simpler approximation. One can e.g. upper bound the expected value of $\mathrm{rk}(\Delta \boldsymbol{A}_i)$ using the Markov bound, cf. (5.4), and c, defined as in Theorem 5.2.5:

$$\mathbb{E}[\mathrm{rk}(\Delta \boldsymbol{A}_i)] \leq \mathbb{E}_{\boldsymbol{A}} = |\mathcal{N}|^2 p_{\Delta\mathcal{N}} \cdot p_w \leq c \cdot \varepsilon_A =: \mathbb{E}_\mathrm{max}. \tag{5.7}$$

We call the upper bound for the equal case $\mathbb{E}_\mathrm{max}$. Due to

$$\mathbb{E}_{\boldsymbol{A}} \leq \mathbb{E}_\mathrm{max} \quad \Longrightarrow \quad \Pr\{\mathrm{rk}(\Delta \boldsymbol{A}_i) > \varepsilon_A\} \leq c,$$

this figure can serve as guideline how the channel deviation is restricted for e.g. a chosen rate region. Therefore its effects are depicted in the following plots.

The statement of Corollary 5.2.6 is illustrated in Figure 5.4. Here the upper bound for the expected values of the random variable of the channel deviation $\mathbb{E}_\mathrm{max}$ is depicted as function of $\hat{\tau}$ for several δ. As usual, for higher δ, there are more opportunities, i.e. $\mathbb{E}_\mathrm{max}$ is larger. The jumps result from the fact, that for each drawn point a respective n and Kullback-Leibler divergences c_1 and c_2 are calculated. Note that here we chose δ, $\tilde{\varepsilon}_\delta = \frac{\varepsilon_\delta}{\tau}$ and $\tilde{\varepsilon}_A = \frac{\varepsilon}{\tau}$ to be fixed for one curve, as in Figure 5.3, but we varied the relation between $\tilde{\varepsilon}_\delta$ and $\tilde{\varepsilon}_A$. Therefore, we get different behavior for equally colored curves (i.e. curves with the same mark). As in Figure 5.3, ε_A is calculated as $\varepsilon_A = \lfloor \tilde{\varepsilon}_A \cdot \lfloor \hat{\tau} \cdot n \rfloor \rfloor$ at each point. Hence we only show the minimum $\underline{\varepsilon}_A$.

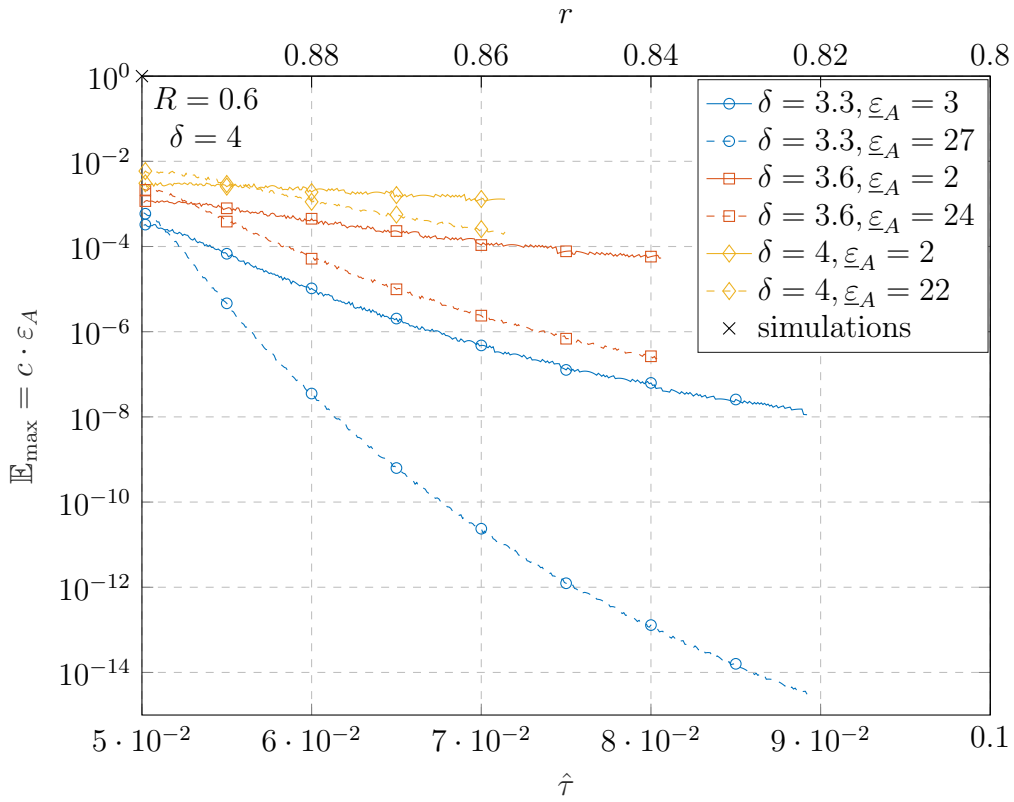


Figure 5.4.: Relation between $\hat{\tau}$ and $\mathbb{E}_{\max}$ for several δ and $\tilde{\varepsilon}_\delta = \varepsilon_\delta/\tau$. $\mathbb{E}_{\max}$ is an upper bound for the expected value of the channel deviation calculated by (5.7). The respective c is displayed in Fig. 5.3. From the variation of $\tilde{\varepsilon}_\delta$ different $\tilde{\varepsilon}_A$ resulted. $\varepsilon_A = \lfloor \tilde{\varepsilon}_A \cdot \lfloor \hat{\tau} \cdot n \rfloor \rfloor$ varies with $\hat{\tau}$, thus we only display the minimum for each curve and denote it by $\underline{\varepsilon}_A$. Parameters: $p_B = 0.05$, $n = n_{\min} + 5$, for the calculation see Fig. 5.3.The step width was chosen to be 0.0001.

Noteworthy is, that for smaller ε_A, the values of $\mathbb{E}_{\max}$ are larger. This means that, according to the Markov approximation, one can allow the channel to vary more, if ε_A is relatively small. Regard, that $\varepsilon_A > 0$ must hold, otherwise the approximation yields $\mathbb{E}_{\max} = 0$. Direct computation of the quantile by $F^{-1}_{\mathrm{rk}(\Delta \boldsymbol{A}_i)}(1-c)$ could give more accurate insight into the behavior. Due to longer computation times, this is not considered here.

Another way to depict the dependencies is to regard $\mathbb{E}_{\max}$ over δ, as shown in Fig. 5.5. Here we also compute the constituents of Corollary 5.2.6 for given δ. As already shown in the previous figure, for larger δ, the expected value is larger too. Fixed parameters are $\hat{\tau} = 0.06$, $p_B = 0.05$ and a step width of 0.001 between calculated points on the abscissa. Since $n_{\min}$ (and therefore $n = n_{\min} + 5$), c and ε_A are calculated for each of these points, one still can see little jumps. Observe, that

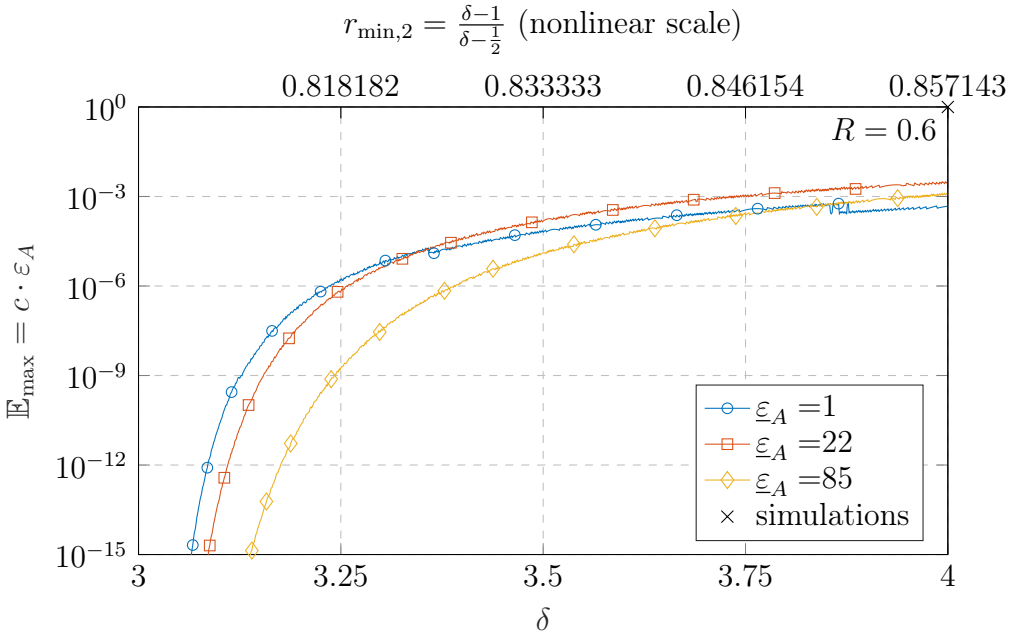


Figure 5.5.: Dependency of $\mathbb{E}_{\max}$ on δ. Bear in mind, that the Markov approximation from (5.7) was used. Parameters: $p_{\mathrm{B}} = 0.05$, $\hat{\tau} = 0.06$, $n = n_{\min} + 5$. c can be looked up in Fig. 5.3 for several values of δ at $\hat{\tau} = 0.06$. $\underline{\varepsilon}_A$ is the minimum of ε_A in each curve. The different manifestations of $\varepsilon_A = \lfloor \tilde{\varepsilon}_A \cdot \lfloor \hat{\tau} \cdot n \rfloor \rfloor$ are obtained by varying $\tilde{\varepsilon}_\delta$. For the calculation of $n_{\min}$, see caption of Fig. 5.3. The plot was generated for a step width of 0.001 between points on the abscissa.

different choices of ε_δ (which cause different $\underline{\varepsilon}_A$) have different effects on certain areas of the curve. Since $\hat{\tau}$ is close to p_{B}, the minimum code length $n_{\min}$ is rather large. On the other hand, the illustration is only possible for $\hat{\tau} < 0.07$ in order to not contradict $r_{\min,2} = 0.857$ on the right side of the plot. $r_{\min,2}$ is calculated via (3.23), recall that this is a lower bound for parameter choices, i.e. the minimum code rate for a specific δ if $n = m$ in the lifted codeword.

5.2.3. Lifting vs. DLNC in Varying Networks

In a similar way as in Theorem 5.2.5, one can use Theorem 4.4.1 as a base for another commensurate statement. This time, all components of the error rank are considered, including the channel deviation, see Figure 5.6. Again the strategy is to find the rate region, where DLNC is better than lifting in a given network, where the channel deviation is specified by the parameters $|\mathcal{N}|$, $p_{\Delta\mathcal{N}}$ and p_w. Detailed analysis and parameter discussion is given in Figures 5.7- 5.9 below the theorem.

$$\operatorname{rk}(\boldsymbol{E}_i) \leq \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) + \operatorname{rk}(\boldsymbol{B}_{i-1}) + \operatorname{rk}(\boldsymbol{B}_i) + \operatorname{rk}(\Delta\boldsymbol{A}_i)$$

Figure 5.6.: Finally the case, where all components of the error rank are considered as they are.

Theorem 5.2.7
Let $R_1, R_2 \sim \operatorname{Bin}(n, p_{\mathrm{B}})$, *A another random variable with the PGF given in* (5.1) , $\alpha = \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\}$, $2np_{\mathrm{B}} < d_{\mathrm{rk,L}}$ *and* $\rho' > 3 - \frac{2}{d_{\mathrm{rk,L}}}$, $c_1 = \mathrm{D}_{\mathrm{KL}}\left(\frac{d_{\mathrm{rk,L}}}{2n}\middle|\middle|p_{\mathrm{B}}\right)$ *and* $c_2 = \min\left\{\mathrm{D}_{\mathrm{KL}}\left(\frac{\left(\frac{\rho'}{2}-1\right)d_{\mathrm{rk,L}}+1}{n}\middle|\middle|p_{\mathrm{B}}\right), c_1 + \frac{\mathrm{e}^{-1}}{\alpha^2}\right\}$ *and* $n > \frac{1}{2(c_1-c_2)}\mathrm{W}_{-1}\left(\alpha^2(c_1-c_2)\right)$ *as in Theorem 4.4.1. Further let* $c \coloneqq \frac{\alpha}{\sqrt{2n}}\mathrm{e}^{-nc_1} - \mathrm{e}^{-nc_2}$, $\varepsilon_A = F_A^{-1}(1-c)$, *where* $F_A^{-1} : [0,1] \to \mathbb{R}$ *is the quantile function of RV A, and* $\rho \geq \rho' + \frac{2\varepsilon_A}{d_{\mathrm{rk,L}}}$. *Then*

$$\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) + A \geq \frac{\rho}{2}d_{\mathrm{rk,L}}\right\} \leq \Pr\left\{2R_2 + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\right\}.$$

Proof:
The proof can be found in Appendix A.4 on p. 88. It combines the procedure of the proofs of Theorem 4.4.1 (comparison of lifting and DLNC for $\operatorname{rk}(\Delta\boldsymbol{A}_i) = 0$) and Theorem 5.2.5, which is concerned with $\operatorname{rk}(\Delta\boldsymbol{A}_i) > 0$, in order to put the statement of Theorem 4.4.1 into the context of varying networks. □

The theorem shows (like the one before) that for given manifestations of the parameters of $\operatorname{rk}(\Delta\boldsymbol{A}_i)$ DLNC is comparably better than lifting, if the error correction capability relates appropriately, in this case, if ρ is large enough. The difference between the two theorems is, that here we chose to compare DLNC to lifting as it is, without lifting taking any advantages.

The combined condition on ρ follows by the conditions $\rho' > 3 - \frac{2}{d_{\mathrm{rk,L}}}$ and $\rho \geq \rho' + \frac{2\varepsilon_A}{d_{\mathrm{rk,L}}}$ from the theorem, so that

$$\rho \geq \rho' + \frac{2\varepsilon_A}{d_{\mathrm{rk,L}}} > 3 - \frac{2}{d_{\mathrm{rk,L}}}(1-\varepsilon_A). \tag{5.8}$$

By the comparison of (4.11) on p. 50 and condition (5.8), we find that here ρ has to be chosen higher than in Theorem 4.4.1 and the discrepancy is mainly depending on the ε_A, i.e. the part defining the variation of $\operatorname{rk}(\Delta\boldsymbol{A}_i)$. Regard that the higher $d_{\mathrm{rk,L}}$, the less impact ε_A has. The higher ε_A, the higher also ρ, but the easier it is to fulfill $\Pr\{\operatorname{rk}(\Delta\boldsymbol{A}_i) > \varepsilon_A\} \leq c$.

In the following we present some figures that contain the outcome of the theorem. For these illustrations the following constraints must hold:

- $\hat{d} > 2p_{\mathrm{B}}$,
- $\frac{1}{n}\left(\left(\frac{\rho}{2}-1\right)d_{\mathrm{rk,L}}-1\right) \leq 1$, since the argument of the Kullback-Leibler divergence must be a p-coin, i.e. $\in [0,1]$,
- the argument of the Lambert-W function must be greater than $-\mathrm{e}^{-1}$ for real values,
- the minimum distance of DLNC must not exceed n, yielding $\rho\hat{d} \leq 1$ and
- $r = 1-\hat{d}+\frac{1}{n}$ must be higher than $r_{\min,2} = \frac{\rho_{\min}+\frac{\rho_{\min}-1}{2\tau}-1}{\rho_{\min}+\frac{\rho_{\min}-1}{2\tau}-\frac{1}{2}}$ (derivarion see (4.12) on p. 51) . Then the statements in the plot hold for $n = m$ and $R_{\mathrm{D}} \geq R_{\mathrm{L}}$.

Note that the figures are restricted most by the last bullet point.

In Figure 5.7 one can see the lower bound c of the difference of the failure probabilities in the static case. For the plot several ρ were chosen and ρ' calculated by $\rho' = 3+\frac{\rho-3}{2}$. Furthermore we chose $p_{\mathrm{B}} = 0.05$, $\alpha = 0.5$ and a step width of 0.0001 between points on the abscissa. In all curves $\underline{\varepsilon}_A$, which is the minimum of ε_A calculated for the plot, resulted to 2. Roughly spoken, for larger $\hat{d}$ (more precisely $\hat{d}$ departing from $2p_{\mathrm{B}}$), respectively lower code rate r, the difference is smaller. Recall that the upper abscissa $r(\infty)$ is a lower bound on the rate for a given $\hat{d}$ and is derived in (4.10) on page 48. Again we see that if n is chosen larger than $n_{\min}$, the jumps get smaller. As before, the jumps can be attributed to the calculation of Kullback-Leibler divergences and resulting n per step.

If the rate region shall be fixed, one can also derive restrictions for the parameters of the channel deviation, as given in the subsequent corollary.

Corollary 5.2.8
Let $R_1, R_2 \sim \mathrm{Bin}(n, p_{\mathrm{B}})$ and define $R_+ \coloneqq R_1 + R_2$, as well as random variables A and L. Let $\rho > 3$, $\rho' \in [3,\rho)$ and $d_{\mathrm{rk,L}} > 2np_{\mathrm{B}}$. Further let $c_1 = \mathrm{D_{KL}}\left(\frac{d_{\mathrm{rk,L}}}{2n}\middle|\middle|p_{\mathrm{B}}\right)$ and $c_2 = \min\left\{\mathrm{D_{KL}}\left(\frac{1}{n}\left(\left(\frac{\rho'}{2}-1\right)d_{\mathrm{rk,L}}+1\right)\middle|\middle|p_{\mathrm{B}}\right), c_1+\frac{\mathrm{e}^{-1}}{\alpha^2}\right\}$, $c = \frac{\alpha}{\sqrt{2n}}\mathrm{e}^{-nc_1}-\mathrm{e}^{-nc_2}$, where $\alpha = \Pr\{L \leq d_{\mathrm{rk,L}}-1\}$ and $n > \frac{1}{2(c_1-c_2)}\mathrm{W}_{-1}(\alpha^2(c_1-c_2))$. With $\varepsilon_A = \frac{d_{\mathrm{rk,L}}}{2}(\rho-\rho')$ and random variable A fulfilling $F_A^{-1}(1-c) \leq \varepsilon_A$ it is

$$\Pr\left\{R_+ + L + A \geq \frac{\rho}{2}d_{\mathrm{rk,L}}\right\} \leq \Pr\left\{2R_2 + L \geq d_{\mathrm{rk,L}}\right\}.$$

Proof:

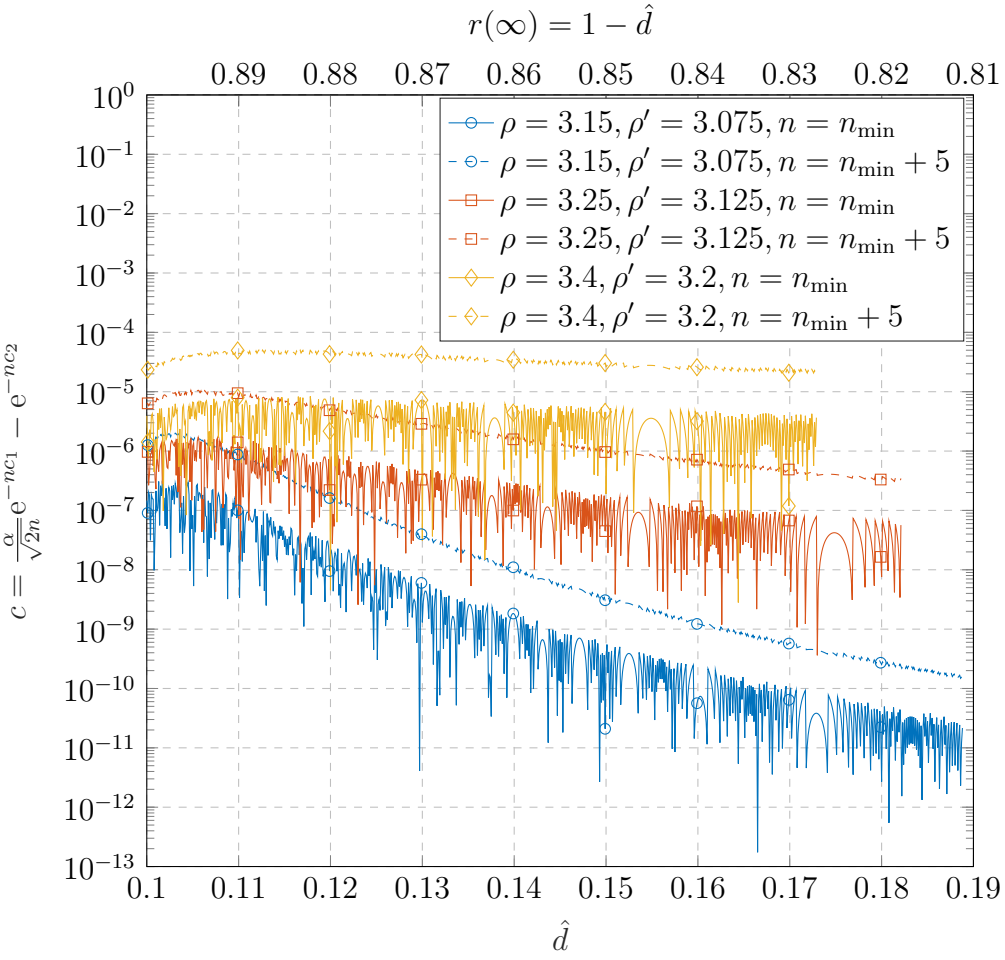


Figure 5.7.: Lower bound for ε according to Theorem 5.2.7 for several ρ. Parameters: $p_{\mathrm{B}} = 0.05$, $\alpha = 0.5$, $\rho' = 3 + (\rho - 3)/2$, $n_{\min} = \lceil \frac{1}{2(c_1 - c_2)} \mathrm{W}_{-1}\left(\alpha^2(c_1 - c_2)\right) \rceil$, $c_1 = \mathrm{D}_{\mathrm{KL}}\left(\hat{d}/2 || p_{\mathrm{B}}\right)$, $c_2 = \mathrm{D}_{\mathrm{KL}}(\hat{d}(\rho'/2 - 1) || p_{\mathrm{B}})$ and step width between points of $\hat{d}$ is 0.0001. The minimum of ε_A is 2 for all curves, where $\varepsilon_A = \lfloor \frac{\rho - \rho'}{2} \lfloor \hat{d} \cdot n \rfloor \rfloor$. $r(\infty)$ from equation (4.10) is a lower bound for the code rate, as explained on page 48.

Since $\varepsilon_A = \frac{d_{\mathrm{rk,L}}}{2}(\rho - \rho')$, it is $\rho = \rho' + \frac{2}{d_{\mathrm{rk,L}}}\varepsilon_A \geq 3 + \frac{2}{d_{\mathrm{rk,L}}}\varepsilon_A > 3 - \frac{2}{d_{\mathrm{rk,L}}}(1 - \varepsilon_A)$. Moreover $F_A^{-1}(1 - c) \leq \varepsilon_A \Leftrightarrow \Pr\{A > \varepsilon_A\} \leq c$. Therefore we have all conditions from Theorem 5.2.7 so that there is nothing left to prove. □

The crucial point of the corollary is that we can decide on a lower bound for the code rate, say $r_{\mathrm{L}} = k_{\mathrm{L}}/n$, where DLNC shall be compared to lifting and find out how much the channel is allowed to vary between generations. Under the presumption $R_{\mathrm{D}} \geq R_{\mathrm{L}}$, the respective ρ can be calculated from the fixed code rate by (4.13) on

p.51. Note that one has to fix τ and f (defined in (3.21) on page 31) beforehand. Having ρ, one can choose $\rho' \in [3, \rho)$, calculate ε_A, c_1, c_2 and finally c and can now variate the parameters of $\mathrm{rk}\,(\Delta \boldsymbol{A}_i)$ to find out, which suffice $F^{-1}_{\mathrm{rk}(\Delta \boldsymbol{A}_i)}(1-c) \leq \varepsilon_A$. Note that $d_{\mathrm{rk,L}}$ (and with it τ) should be fixed in the beginning, but since we only have a lower bound on n, this does not necessarily impair the rate.

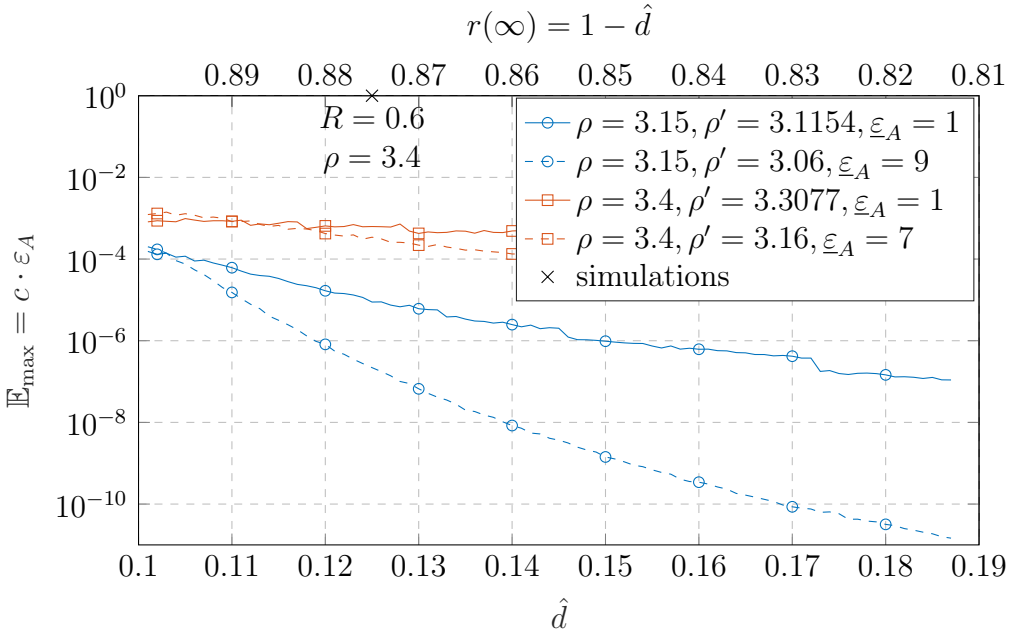


Figure 5.8.: Relation between the upper bound of the expected values of the RV of the channel deviation for specified ρ and ρ'. ε_A can be calculated from ρ and ρ' by $\varepsilon_A = \frac{d_{\mathrm{rk,L}}}{2}(\rho - \rho')$, here we calculated n for each step, therefore $d_{\mathrm{rk,L}} = \lfloor \hat{d} \cdot n \rfloor$. Note that the Markov approximation was used for the calculation of $\mathbb{E}_{\max}$. Parameters: $p_{\mathrm{B}} = 0.05$, $\alpha = 0.5$, , $n = n_{\min} + 5$ (calculation of $n_{\min}$ and c see Fig. 5.7) and a step width of 0.001. ε_A varied with $\hat{d}$, therefore we only print the minimum $\underline{\varepsilon}_A$.

Using the Markov inequality (5.4), one can calculate the bound $\mathbb{E}_{\max} = c \cdot \varepsilon_A$. Here c is the lower bound of the failure probabilities in Theorem 5.2.7. In Figure 5.8 we chose $p_{\mathrm{B}} = 0.05$, $\alpha = 0.5$, $n = n_{\min} + 5$ and a step width of 0.001 to display the relation to $\hat{d}$. For each $\hat{d}$ we calculated $c_1, c_2, n_{\min}$ and c for specified ρ and ρ', as given in Corollary 5.2.8. The upper bound of the $(1-c)$-quantile of RV A is calculated via

$$\varepsilon_A = \left\lfloor \frac{\rho - \rho'}{2} \lfloor \hat{d} \cdot n \rfloor \right\rfloor . \tag{5.9}$$

Since ε_A varied with $\hat{d}$, we only display the minimum $\underline{\varepsilon}_A$ for each curve. As expected we get larger $\mathbb{E}_{\max}$ for higher choices of ρ. On the other hand by increasing also ρ', which leads to relatively small ε_A, the curve also rises.

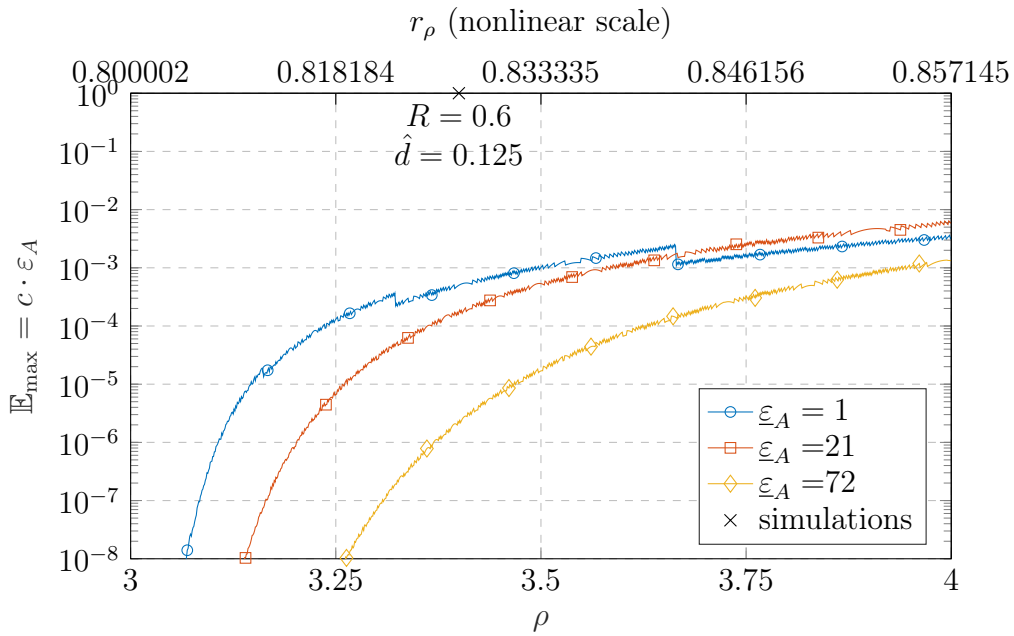


Figure 5.9.: The upper bound $\mathbb{E}_{\max}$ as a function of ρ, respectively the lower bounded minimum code rate r_ρ from (4.12) on p. 51. $\rho' \in (3, \rho)$ is varied, from ρ and ρ' the quantile ε_A is calculated for every point in the plot, cf. (5.9), therefore only the minimum $\underline{\varepsilon}_A$ is depicted. Parameters: $p_{\mathrm{B}} = 0.05$, $\hat{d} = 0.125$, $\alpha = 0.5$, $n = n_{\min} + 5$ and a step width of 0.001. The calculation of c and $n_{\min}$ is given in Fig. 5.7, approximate values for c can be read of the mentioned figure for $\hat{d} = 0.125$ and several ρ.

The bound $\mathbb{E}_{\max}$ can also be regarded as function of ρ. This is portrayed in Figure 5.9. The upper abscissa r_ρ is calculated as in (4.12). In the plot we have $p_{\mathrm{B}} = 0.05$, $\hat{d} = 0.125$, $\alpha = 0.5$, $n = n_{\min} + 5$ and a step width of 0.001. ε_A is calculated for every point in the plot, therefore only the minimum $\underline{\varepsilon}_A$ is depicted. Note that the simulations were executed above the curves, although $\hat{d}$ coincides. One can see that the larger ε_A, the smaller $\mathbb{E}_{\max}$, but this does not necessarily hold for all regions of the picture. It is important to note, that the calculated $n_{\min}$ are rather large, because of the proximity of $\hat{d}$ to $2p_{\mathrm{B}}$. $\hat{d}$ must be chosen carefully for the depicted window, since otherwise it can contradict the lower bound on the code rate given by $r_{\min,2}$. Note that there exist better bounds than the Markov bound. Using tighter bounds, as e.g. the Chernoff bound, would yield better approximations for the parameters.

6. Simulation

After the theoretical analysis, we proceed with more practical comparisons. We define a block error as the incidence that in one generation a receive matrix can not be decoded. By counting the number of decoding failures for a high number of generations, we can calculate the Block Error Rate (BLER) for each scenario. The number of generations that are carried out with the same starting channel matrix is chosen to be 10. Since we have only a small number of generations, which are dependent on each other, this procedure is a usual Monte-Carlo Simulation. In each execution of a simulation a transmit matrix is generated for the respective mode, these are `Lifting` or `DLNC`. In case of `Lifting`, the all zero codeword can be used throughout all simulations, which saves the time of encoding in the Gabidulin code. For the `DLNC` case it is not possible to use the same codeword for all simulations, since the error depends on the codeword. Therefore random matrices are drawn and encoded. Subsequently, the transmit matrix is sent via the channel. Its model is implemented as described in Section 6.1. After the transmission it is checked, whether decoding is possible.

6.1. Structure of the Implemented Channel Model

The simulation functions and scripts were implemented using SageMath [S+18], a free open-source mathematics software system licensed under the GNU General Public License. Further a library for coding theory in SageMath named "codinglib" [Ros] by Johan Rosenkilde was used for parallel computing of the simulations and as base for a library on Gabidulin codes that was provided by Sven Puchinger. The implementation allows computations in $\mathbb{F}_{q^m}$ for q power of a prime and also $m \in \mathbb{N}$ and arbitrary code lengths $n \in \mathbb{N}$.[1]

All functions necessary for the channel were collected in a folder called `rlnclib`. There one can find the following functions:

`A = init_rlnc_channel(Fq,N,n,rkA='x')` for the initialization of the RLNC chan-

[1] The only exception is the computation of the pseudoinverse for DLNC, where $q = 2$ and $m > 1$ must hold, due to its implementation in SageMath.

nel, i.e. the generation of a channel matrix `A` over `Fq` of desired dimensions `N`×`n`. It is possible to specify, whether `A` shall have full rank or not. `x` means arbitrary, `n` gives full rank.

`(Y,A) = rlnc_channel(A,X,p_B,numNodes,p_L,p_W,rkDelta=0,[...])` channel model for RLNC, where `Y`= `A`$\cdot$ `X` $+\boldsymbol{B}$ with $\mathrm{rk}\,(\boldsymbol{B}) \sim \mathrm{Bin}$(`n`,`p_B`). `rkDelta` decides whether `A` is changed before the calculation. If so, a number ℓ is randomly chosen according to $L \sim \mathrm{Bin}$(`numNodes`,`p_L`) an then ℓ rank weights w_i, $i \in \{1, \ldots, \ell\}$ are chosen from the distribution Bin(`numNodes`,`p_w`) and summed up. The resulting rank weight is then imprinted in $\Delta\boldsymbol{A}_i$, which is chosen randomly with the constraint of the given rank. For more information on properties of $\mathrm{rk}\,(\Delta\boldsymbol{A}_i)$ and its distribution see Section 2.4.1 on p. 18.

`X = lifting(S)` realizes the lifting construction. `X` $= [\, \boldsymbol{I} \mid$ `S` $\,]$ is returned, where $\boldsymbol{I}$ is an identity matrix of size $n \times n$, if n is the first dimension of `S`.

`X_i = dlnc_modulation(X_imin1,S_i)` modulates the information `S_i` on to the previous state matrix `X_imin1` by ordinary matrix multiplication, resulting in the current transmit matrix `X_i`, i.e. `X_i = X_imin1[:,0:n]*S_i`, where `n` is the first dimension of `S_i`, cf. Section 2.4.

`Shat = dlnc_demodulation(Y_i,Y_imin1)` implements the demodulation step of DLNC according to [SCFH13], which is executed via

```
S_hat = weakpseudoinverse(Y_imin1[0:n,0:n]) * Y_i,
```

where `n` is the first dimension of `Y_i`.

`Y_inv = weakpseudoinverse(Y)` contains the calculation of the weak pseudoinverse, used for the demodulation of DLNC. The implementation of the algorithm given in 6.3.1.

`is_decodable(mode,*args)` is a function checking if the decoding guarantee is given for several `mode`s, so far implemented are `DLNC` and `Lifting`. For more informations on the check see Sections 6.2 and 6.3.

and some helper functions. The implementation of these functions is displayed in Appendix B. During the execution of the simulations, these functions were called in a function named `simulate_rlnc(...)`, which generates the transmit matrices, carries out transmissions for `numGen` generations in a row for a certain `mode` and returns the number of decoding failures.

6.2. Decoding Guarantee for Lifted Gabidulin Codes

The decoding step is omitted by checking if decoding can be guaranteed. Ensuing Section 2.3.3 decoding is possible, if (2.13), see page 15, is fulfilled, i.e. if

$$d_{\mathrm{s}}(\langle \boldsymbol{X} \rangle, \langle \boldsymbol{Y} \rangle) = \mathrm{rk} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} - \mathrm{rk}\,(\boldsymbol{X}) - \mathrm{rk}\,(\boldsymbol{Y}) \leq d_{\mathrm{rk}} - 1.$$

In the simulation environment $\boldsymbol{X}$ and $\boldsymbol{Y}$ are produced, so that the decoding condition can be verified easily. d_{rk} is the minimum distance of the lifted Gabidulin code.

6.3. Decoding Guarantee for Gabidulin Codes in DLNC

For DLNC we use the demodulation introduced by [SCFH13], see also Section 2.4.2, and, according to Theorem 2.4.3, we check if

$$\mathrm{rk}\left(\hat{\boldsymbol{S}}_i - \boldsymbol{S}_i\right) \leq \left\lfloor \frac{d_{\mathrm{rk}} - 1}{2} \right\rfloor,$$

where d_{rk} is the minimum distance of the applied Gabidulin code. $\hat{\boldsymbol{S}}_i$ is calculated as given in equation (2.18) on p. 17. The demodulation is based on the pseudoinverse of the receive matrix.

6.3.1. Calculation of the Pseudoinverse for DLNC

Let $\boldsymbol{Y}_i$ be the received matrix (or in case it is not square of the front square part of the received matrix) in generation i. Let $\boldsymbol{Y}_i \in \mathbb{F}_q^{n \times n}$. Then its pseudoinverse $\boldsymbol{Y}_i^+$ must have full rank and fulfill

$$\boldsymbol{Y}_i^+ \cdot \boldsymbol{Y}_i = \boldsymbol{I} + \boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}, \tag{6.1}$$

where $\mathcal{U}$ is the set consisting of the positions $p_i \in \{0, \ldots, n\}$ that are no leading positions in the reduced row echelon form (RRE) of $\boldsymbol{Y}_i^+ \cdot \boldsymbol{Y}_i$ and $\boldsymbol{L}$ is a matrix of appropriate dimensions, i.e. $n \times |\mathcal{U}|$. All in all the product looks like an identity matrix containing some arbitrary columns.

In SageMath the built-in function `mat.inverse()`, where `mat` is a matrix object, is suited for the computation of the desired pseudomatrix. For fields of characteristic 2 the function returns a full-rank matrix even for non-invertible matrices. A check of the properties yields, that (if at all) only some rows have to be switched to get (6.1).

6.4. Simulation Results

This section comprises the results of the simulations, where lifting and DLNC are compared in several scenarios, i.e. in static or slowly-varying networks and for different rates. For one simulation scenario (next to other parameters) an overall rate R is chosen and the dimensions of the codes computed accordingly, cf. (3.11) and (3.13) on p. 28. The code used for lifting has a dimension of

$$k_{\mathrm{L}} = R \cdot n \cdot \frac{n+m}{m}. \tag{6.2}$$

The dimensions for DLNC were chosen by $k_{\mathrm{D}} = \lfloor \frac{R \cdot n}{1 - \frac{n}{q(n+m)}} \rfloor$, although this contradicts $R_{\mathrm{D}} \geq R_{\mathrm{L}}$, because the difference between both overall rates is $|R_{\mathrm{L}} - R_{\mathrm{D}}| < 0.001$ for the chosen parameters. These were $q = 256$, $n = 40$, as in [SCFH13], and $m = 80$, which is a little smaller than it is chosen in [SCFH13].

6.4.1. Static Networks

Here we regard only channel deviations of zero rank, i.e. $\mathrm{rk}\,(\Delta \boldsymbol{A}_i) = 0$. We regarded two different overall rates. The first one is $R_{\mathrm{L}} = 0.45$, which leads to $k_{\mathrm{L}} = 27$ and $k_{\mathrm{D}} = 18$. Hence we have a relation of

$$\delta_{\mathrm{sim}} = \frac{\tau_{\mathrm{D}}}{\tau_{\mathrm{L}}} = \frac{\lfloor \frac{n-k_{\mathrm{D}}}{2} \rfloor}{\lfloor \frac{n-k_{\mathrm{L}}}{2} \rfloor} = 1.8\bar{3} \tag{6.3}$$

between both error correction radii. Due to $n = 40$ and $k_{\mathrm{L}} = 27$, we are regarding $\hat{d} = \frac{n-k_{\mathrm{L}}}{n} = 0.325$. Theorem 4.4.1 helps to find, that for $\rho = 3.15$ (which corresponds in our case to $\delta \approx 3.33$), we have $n_{\min} = 40$. Note that this combination of ρ and $\hat{d}$ does not offer the possibility to compare lifting and DLNC, because the minimum distance $d_{\mathrm{rk,D}} = \rho(n - k_{\mathrm{L}} + 1)$ would be larger than n.

However with $\delta = 3.33$ the minimum overall rate is

$$R_{\min} = \frac{m}{n+m} r_{\min,f} \underset{(3.21)}{\overset{(3.22)}{=}} \frac{2}{3} \frac{\delta - 1}{\delta - \frac{m}{M\left(1 - \frac{n}{qM}\right)}} \overset{\delta = 3.33}{\approx} 0.5833. \tag{6.4}$$

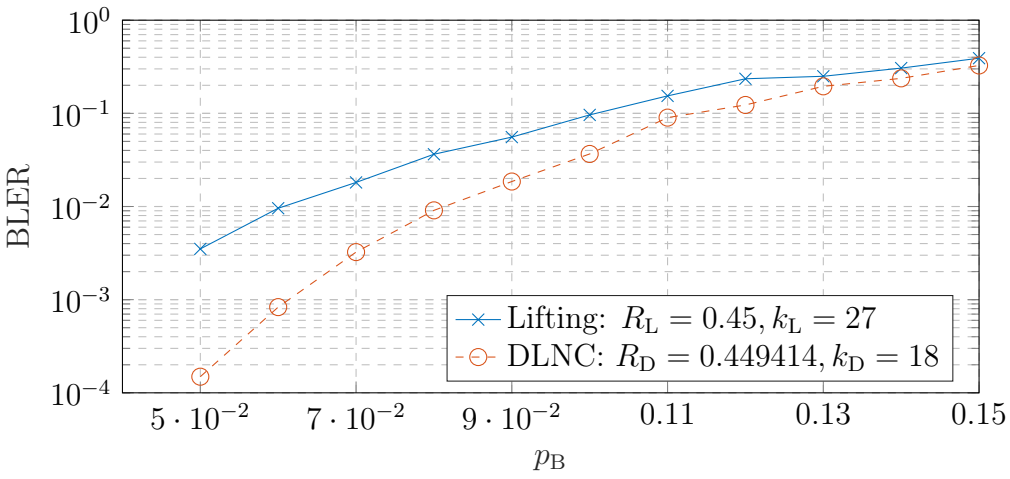


Figure 6.1.: Simulation result for the first scenario with $R = 0.45, m = 80$ in a static network. The number of different simulated channel matrices varied between 80 and 26789. At least 40 decoding failures were simulated. Note that the dimensions for DLNC was chosen to be $k_{\mathrm{D}} = \lfloor \frac{R \cdot n}{1 - \frac{n}{q(n+m)}} \rfloor$ because the difference between both overall rates is $|R_{\mathrm{L}} - R_{\mathrm{D}}| < 0.001$.

The maximum rate, where the lifted Gabidulin code can be used, is due to the parameter choices

$$R_{\max} \overset{(3.12)}{=} \frac{m}{n+m} = \frac{2}{3} = 0.\bar{6}. \tag{6.5}$$

This reveals, that the overall rate we chose, is smaller than the rate, where the statements from Chapter 4 hold. As it can be seen in Figure 6.1, DLNC performs better nevertheless.

The second rate, that was chosen for simulations, is $R = 0.6$. The outcome is shown in Figure 6.2. As we can see by comparing Figure 6.1 and 6.2, for higher rates, the error performance of DLNC differs even more from the lifting procedure, i.e. it is visible that the relation between the BLER of DLNC and lifting is larger for $R = 0.6$ and $p_{\mathrm{B}} \leq 0.1$. Nevertheless, one can also see that the performance itself is worse than in Fig. 6.1, due to the higher rate. Having $k_{\mathrm{L}} = 36$ (due to (6.2))and $k_{\mathrm{D}} = 24$, we get $\delta_{\mathrm{sim}} = 4$, for the calculation see (6.3). The theoretical δ, that emanates from the choice of $k_{\mathrm{L}} = 36$ and $n = 40$ is $\delta \approx 5.41$. Here the choices $\rho = 4.53$ and $\hat{d} = 0.1$ do not contradict the definition of the minimum distance $d_{\mathrm{rk,D}}$. The minimum overall rate for the chosen parameters yields a rate region

$$R \geq R_{\min} \underset{\delta=5.41}{\overset{(6.4)}{\approx}} 0.62,$$

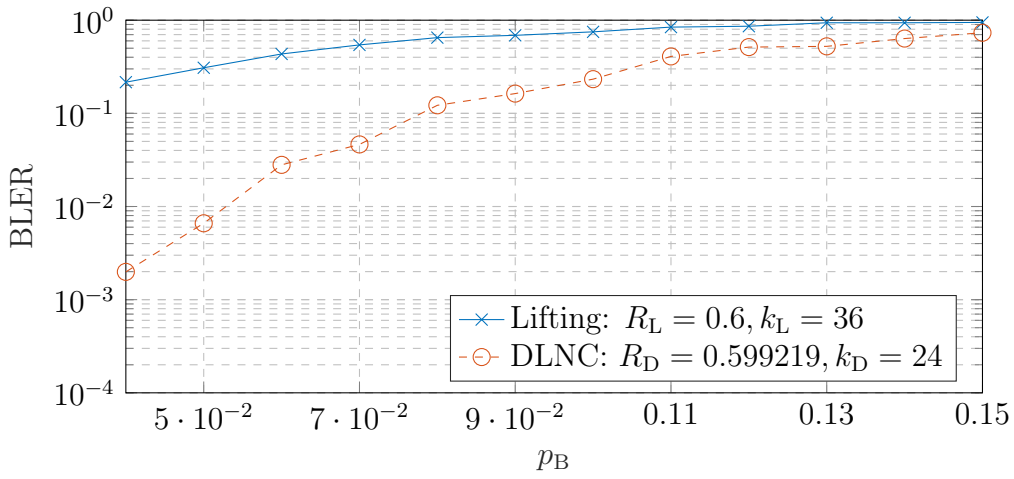


Figure 6.2.: Simulation result for a static network and parameters $q = 256$, $n = 40$, $m = 80$. The number of different simulated channel matrices varied between 15 and 4879. At least 58 decoding failures were simulated. Note that the dimensions for DLNC was chosen by $k_D = \lfloor \frac{R \cdot n}{1 - \frac{n}{q(n+m)}} \rfloor$ because the difference between both overall rates is $|R_L - R_D| < 0.001$.

which is again higher than the simulated overall rate. The upper bound is $R_{\max}$ as given in (6.5).

6.4.2. Slowly-Varying Networks

For the simulation of slowly-varying networks we fixed the parameters $|\mathcal{N}|$, p_w and $p_{\Delta\mathcal{N}}$, as introduced in Section 2.4.1, additionally to the already determined parameters $q = 256$, $n = 40$ and $m = 80$. Again we regarded the two overall rates $R = 0.45$ and $R = 0.6$ from before.

We chose $|\mathcal{N}| = 1000$, and $p_w = 0.01$ for both rates. For the number of nodes we set a restriction $|\mathcal{N}| > 3n$, because transmitter and receiver are connected to at least n nodes and there should be at least n nodes in between. Since p_w reflects the interconnection of the network and we know that transmitter and receiver are connected to n nodes we choose p_w near the relation $\frac{n}{|\mathcal{N}|} = \frac{40}{1000}$.

For the simulations with rate $R = 0.45$ we found that for $p_{\Delta\mathcal{N}} = 10^{-5}$ the performance of the DLNC scheme leaves lifting behind. This choice of $p_{\Delta\mathcal{N}}$ leads, together with the former parameter determinations, to an upper bound of the expected value of the rank of the channel deviation $\mathbb{E}_{\boldsymbol{A}} = \mathbb{E}\left[\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)}\right] = 0.1$, which means that

in, on average, every tenth generation there is a change in the rank of one, i.e. $\mathrm{rk}\,(\Delta \boldsymbol{A}_i) = 1$. The results are shown in Figure 6.3.

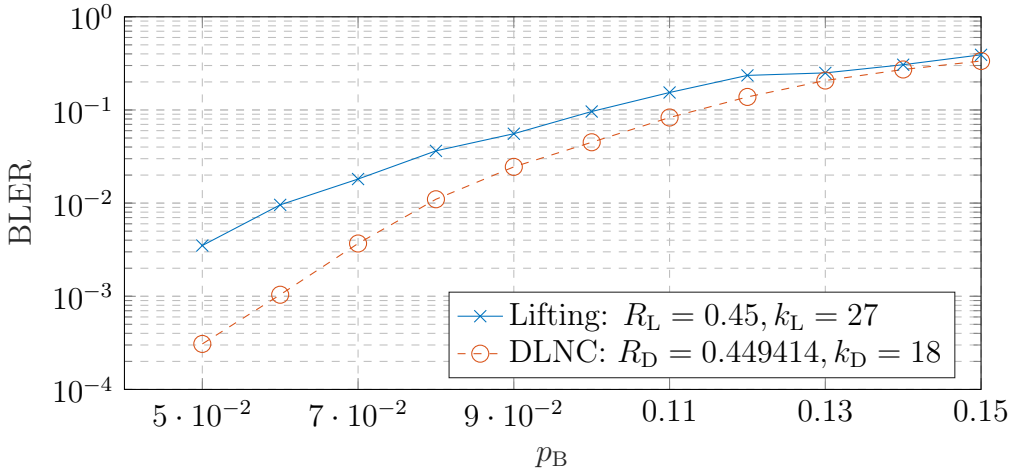


Figure 6.3.: Varying network simulation result for $q = 256$, $n = 40$, $m = 80$. The number of different simulated channel matrices varied between 50 and 15608. At least 40 decoding failures were simulated. The expected value of the rank of the channel deviation in this figure is $\mathbb{E}\left[\overline{\mathrm{rk}\,(\Delta \boldsymbol{A}_i)}\right] = |\mathcal{N}|^2 \cdot p_{\Delta n} \cdot p_w = 0.1$ with $|\mathcal{N}| = 1000$, $p_{\Delta\mathcal{N}} = 0.00001$ and $p_w = 0.01$. Note that the dimensions for DLNC is $k_{\mathrm{D}} = \left\lfloor \frac{R \cdot n}{1 - \frac{n}{q(n+m)}} \right\rfloor$ because the difference between both overall rates is $|R_{\mathrm{L}} - R_{\mathrm{D}}| < 0.001$.

The other rate, $R = 0.6$, is simulated for $p_{\Delta\mathcal{N}} = 10^{-4}$, then we have $\mathbb{E}_{\boldsymbol{A}} = 1$. As shown in Figure 5.8 and 5.9, this parameter choice is a point beyond the bounds derived in Chapter 5, i.e. the theoretical analysis could not ensure that DLNC is better than lifting for these parameters. In Figure 6.4 one can see the performance of both schemes. Again it is shown, that the overall performance suffers due to the higher rate, compared to $R = 0.45$. However we have a gain in the error correction capability by using DLNC instead of lifting even for the higher choice of the expected value for the channel deviation $\mathbb{E}_{\boldsymbol{A}}$.

Comparing Figures 6.1 and 6.3 or respectively Fig. 6.2 and 6.4 shows that, as expected, the overall performance of the DLNC procedure worsens when channel variations occur, but nevertheless outperforms lifting in these kinds of slowly-varying networks.

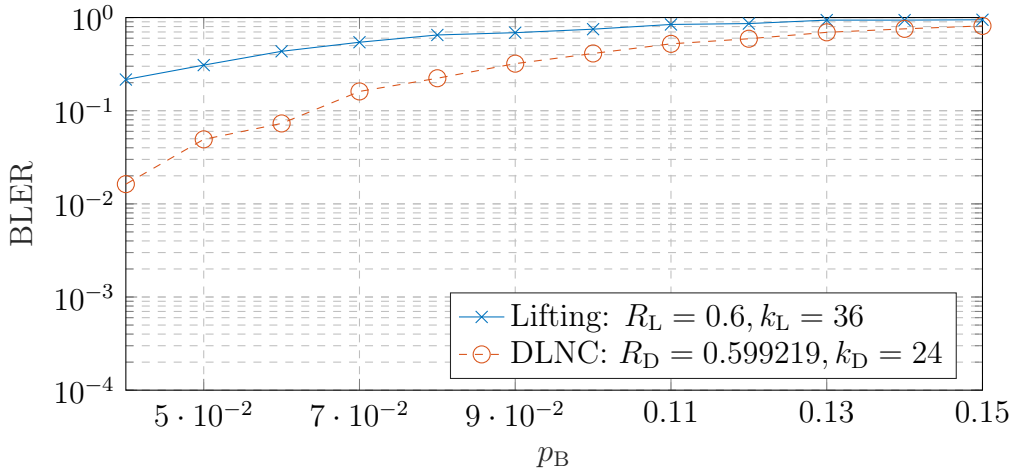


Figure 6.4.: Simulation result for a varying network and parameters $q = 256$, $n = 40$, $m = 80$. The number of different simulated channel matrices varied between 68 and 725. At least 64 decoding failures were simulated. The expected value of the rank of the channel deviation in this figure is $\mathbb{E}_{\boldsymbol{A}} = \mathbb{E}\left[\mathrm{rk}\left(\Delta \boldsymbol{A}_i\right)\right] = |\mathcal{N}|^2 \cdot p_w \cdot p_{\Delta\mathcal{N}} = 1$ with $|\mathcal{N}| = 1000$, $p_{\Delta\mathcal{N}} = 0.0001$ and $p_w = 0.01$. Note that the dimensions for DLNC was chosen as $k_D = \lfloor \frac{R \cdot n}{1 - \frac{n}{q(n+m)}} \rfloor$ because the difference between both overall rates is $|R_L - R_D| < 0.001$

7. Conclusion

In this work, after introducing the main concepts of rank-metric codes, Random Linear Network Coding (RLNC), lifting and Differential Linear Network Coding (DLNC) in Chapter 2, a probabilistic analysis of the procedures DLNC and lifting has been carried out by comparing their failure probabilities in several steps in Chapter 4. We started to consider an intuitive view on the scheme with double error rank and double error correction radius for DLNC in static networks in Section 4.2. In the subsequent section, a rough estimate for the error rank in a static network was derived and an analog comparison conducted. In a last step in Section 4.4, the tightest possible approximations were used to estimate the relation between the failure probabilities of lifting and DLNC. The outcome of these comparisons were lower bounds on the code length n, which means that for long enough codes DLNC will outperform lifting in a certain rate region, if the network is static.

Facing varying networks in Chapter 5, we could reuse the results gained for static networks and derive bounds for the variations, i.e. the parameters of the channel deviation for a given rate region and vice versa. Furthermore, we introduced a definition for slowly-varying channels and examined the restrictions for networks, where DLNC shall give an advantage over lifting.

The simulations in Chapter 6 confirm the theoretical analysis, showing that DLNC surpasses lifting even beyond the derived parameter bounds. This is due to the approximations that were necessary for the comparison. Note that for the simulations, we made approximations that worsen the scenario, so it can be expected that in reality the procedures (and particularly DLNC in the varying case) perform better.

This work offers a variety of starting points for further research. This could involve investigations of full-rank Gabidulin codewords and their construction in order to find a way to map information on codewords for the usage in a DLNC scenario. Moreover, as suggested in [PCF+15], the usage of Partial Unit Memory (PUM) codes could be combined with DLNC to gain better performance. Since (P)UM codes are special convolutional codes constructed from block codes, they are particularly suited to correct error peaks in a sequence of errors. Such error peaks occur in DLNC if the network changes in only a few generations. Therefore, (P)UM codes might admit a larger network variation compared to DLNC with ordinary rank-

metric codes, and also compared to (P)UM codes in a lifting construction. Similarly, interleaved Gabidulin codes using DLNC could be compared to lifted interleaved Gabidulin codes of the same size, as indicated in Section 3.3. Since Error Trapping in combination with channel sounding exceeds lifted Gabidulin codes for large field sizes according to [Cyr17, p. 106], it would be interesting to see how DLNC and Error Trapping perform relative to each other.

Further bounds could be derived for the parameters of the channel deviation or its quantiles by using e.g. the derived Probability Generating Function in addition to the Chernoff bound.

In conclusion, DLNC seems to be a suitable improvement to the lifting construction in slowly-varying networks, where RLNC is applied.

Acronyms

AMC	Additive Matrix Channel
BLER	Block Error Rate
CDF	Cumulative Density Function
DLNC	Differential Linear Network Coding
DPSK	Differential Phase-Shift Keying
MAMC	Multiplicative Additive Matrix Channel
MDS	Maximum Distance Separable
MGF	Moment Generating Function
MRD	Maximum Rank Distance
NC	Network Coding
PDF	Probability Density Function
PGF	Probability Generating Function
PMF	Probability Mass Function
PSK	Phase-Shift Keying
PUM	Partial Unit Memory
RLNC	Random Linear Network Coding
RRDM	Receive Rank Deficiency Matrix
RRE	reduced row echelon form
RS	Reed Solomon
RV	random variable

Notations

$\mathrm{rk}\,(\cdot)$	Rank of a Matrix
$\langle \cdot \rangle$	Row Space of a Matrix
$\left[\begin{smallmatrix} \cdot \\ \cdot \end{smallmatrix} \right]_q$	Gaussian coefficient
q	Power of a prime
$\mathbb{F}_q$	Finite field of order q
$\mathbb{F}_{q^m}$	Extension field of $\mathbb{F}_q$ of degree m
$\mathbb{F}_{q^m}^n$	Set of all row vectors over $\mathbb{F}_{q^m}$ of length n
$\mathbb{F}_{q^m}^{s \times n}$	Set of all $s \times n$ matrices over $\mathbb{F}_{q^m}$
n	Number of transmitted packets in each generation
N	Number of received packets in each generation
$M = n + m$	Second dimension of sending matrix, m is the extension degree of the Lifted Gabidulin Code
$\mathrm{GL}_n(\mathbb{F}_q)$	Set of invertible $n \times n$ matrices in $\mathbb{F}_q$ or general linear group
$\Pr\{A\}$	Probability of event A
$\mathbb{E}[X]$	Expected value of random variable X
$X \sim \mathrm{Bin}\,(n, p)$	Random variable X is binomially distributed with n the number of trials and p the probability of a success
$X \sim \mathrm{Pois}\,(\lambda)$	Random variable X is Poisson distributed with with parameter λ as success rate
$\ln(x)$	Natural logarithm of x with basis e
R	Overall rate of a transmission scheme
$r = \frac{k}{n}$	Code rate

A. Proofs

Here we present the remaining proofs from Chapter 4 and 5.

A.1. Proof of Theorem 4.3.3

Before proving Theorem 4.3.3, we state the theorem again.

Theorem 4.3.3
Let $R_1, R_2 \sim \mathrm{Bin}\,(n, p_\mathrm{B})$. *Further* $\delta > 3$, $0 < \varepsilon_\delta \leq (\delta - 3)\tau/3$, $c_1 \coloneqq \mathrm{D}_\mathrm{KL}\left(\frac{\tau}{n}||p_\mathrm{B}\right)$ *and* $c_2 \coloneqq \min\{\mathrm{D}_\mathrm{KL}\left(\frac{\tau+\varepsilon_\delta}{n}||p_\mathrm{B}\right), c_1 + 4\mathrm{e}^{-1}\}$.[1] *If* $n > \frac{\mathrm{W}_{-1}((c_1-c_2)/4)}{2(c_1-c_2)}$, *then*

$$\Pr\{R_1 + 2R_2 \geq \delta\tau\} < \Pr\{R_2 \geq \tau\}.$$

Let R_1 and R_2 be the desired random variables, i.e. $R_1, R_2 \sim \mathrm{Bin}\,(n, p_\mathrm{B})$. Let further $0 < \varepsilon_\delta < (\delta - 3)\frac{\tau}{3}$, $c_1 \coloneqq \mathrm{D}_\mathrm{KL}\left(\frac{\tau}{n}||p_\mathrm{B}\right)$, $c_2 \coloneqq \min\{\mathrm{D}_\mathrm{KL}\left(\frac{\tau+\varepsilon_\delta}{n}||p_\mathrm{B}\right), c_1 + 4\mathrm{e}^{-1}\}$ and $n > \frac{\mathrm{W}_{-1}((c_1-c_2)/4)}{2(c_1-c_2)}$ as demanded in the theorem. By the law of total probability, we get

$$\begin{aligned}\Pr\{R_1 + 2R_2 \geq \delta\tau\} &= \sum_{i=0}^{n} \Pr\{R_1 = i\} \Pr\{2R_2 + R_1 \geq \delta\tau | R_1 = i\} \\ &= \sum_{i=0}^{n} \Pr\{R_1 = i\} \Pr\{R_2 \geq (\delta\tau - i)/2\}.\end{aligned}$$

Regard the following partition

$$\begin{aligned}\Pr\{R_1 + 2R_2 \geq \delta\tau\} = \sum_{i=0}^{\tau+\varepsilon_\delta} \Pr\{R_1 = i\} \Pr\left\{R_2 \geq \frac{\delta\tau - i}{2}\right\} + \\ + \sum_{i>\tau+\varepsilon_\delta} \Pr\{R_1 = i\} \Pr\left\{R_2 \geq \frac{\delta\tau - i}{2}\right\}.\end{aligned} \tag{A.1}$$

[1] The last definition guarantees $c_2 - c_1 \leq 4\mathrm{e}^{-1}$.

Then we can find an upper bound for the second addend by

$$\sum_{i>\tau+\varepsilon_\delta} \Pr\{R_1 = i\} \underbrace{\Pr\left\{R_2 \geq \frac{\delta\tau - i}{2}\right\}}_{\leq 1} \leq \Pr\{R_1 \geq \tau + \varepsilon_\delta\}.$$

Since we assume $\varepsilon_\delta < (\delta - 3)\frac{\tau}{3}$, we get

$$\begin{aligned} & \varepsilon_\delta \leq (\delta - 3)\frac{\tau}{3} && \text{(A.2)} \\ \iff \quad & 3\varepsilon_\delta \leq (\delta - 3)\tau \\ \iff \quad & \delta\tau - \tau - \varepsilon_\delta \geq 2(\tau + \varepsilon_\delta) \\ \iff \quad & \frac{\delta\tau - \tau - \varepsilon_\delta}{2} \geq \tau + \varepsilon_\delta, && \text{(A.3)} \end{aligned}$$

In the sum of (A.1) it is

$$\Pr\left\{R_2 \geq \frac{\delta\tau - i}{2}\right\} \leq \Pr\left\{R_2 \geq \frac{\delta\tau - \tau - \varepsilon_\delta}{2}\right\} \quad \forall i \leq \tau + \varepsilon_\delta, \tag{A.4}$$

therefore we get

$$\begin{aligned} \Pr\{R_1 + 2R_2 > \delta\tau\} &\leq \sum_{i=0}^{\tau+\varepsilon_\delta} \Pr\{R_1 = i\} \Pr\left\{R_2 \geq \frac{\delta\tau - i}{2}\right\} \\ &\overset{\text{(A.4)}}{\leq} \Pr\left\{R_2 \geq \underbrace{\frac{\delta\tau - \tau - \varepsilon_\delta}{2}}_{\overset{\text{(A.3)}}{\geq} \tau+\varepsilon_\delta}\right\} \underbrace{\sum_{i=0}^{\tau+\varepsilon_\delta} \Pr\{R_1 = i\}}_{\leq 1} + \Pr\{R_1 \geq \tau + \varepsilon_\delta\} \\ &\leq \Pr\{R_1 \geq \tau + \varepsilon_\delta\} + \Pr\{R_2 \geq \tau + \varepsilon_\delta\}. \end{aligned}$$

Since R_1 and R_2 are independent and identically binomially distributed, we can use the tailbound ($<$) for both random variables, yielding

$$\Pr\{R_1 + 2R_2 \geq \delta\tau\} \leq 2\exp\left\{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{\tau + \varepsilon_\delta}{n}\middle|\middle|p_{\mathrm{B}}\right)\right\}.$$

Lemma 4.3.2 can be used, since all conditions are fulfilled, where we define $\alpha = \frac{1}{2}$ and $np_{\mathrm{B}} < x_1 \coloneqq \tau < x_2 \coloneqq \tau + \varepsilon_\delta$. Then the result follows by

$$\begin{aligned} \Pr\{R_1 + 2R_2 \geq \delta\tau\} &\leq 2\exp\left\{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{\tau + \varepsilon_\delta}{n}\middle|\middle|p_{\mathrm{B}}\right)\right\} \\ &< 2\frac{1}{2}\frac{1}{\sqrt{2n}}\exp\left\{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n}\middle|\middle|p_{\mathrm{B}}\right)\right\} \\ &\overset{(>)}{\leq} \Pr\{R_2 \geq \tau\} \end{aligned}$$

for

$$n > \frac{\mathrm{W}_{-1}(\alpha^2(c_1 - c_2))}{2(c_1 - c_2)} = \frac{\mathrm{W}_{-1}((c_1 - c_2)/4)}{2(c_1 - c_2)}.$$

□

A.2. Proof of Theorem 4.4.1

For better readability we restate the theorem here.

Theorem 4.4.1
Let $R_+ = R_1 + R_2$ be the sum of two binomially distributed random variables, i.e. $R_1, R_2 \sim \mathrm{Bin}(n, p_\mathrm{B})$, where $np_\mathrm{B} < \frac{d_{\mathrm{rk,L}}}{2} < \left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1$. Let $\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right)$ be another random variable and $\alpha = \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\}$. Further $c_1 \coloneqq \mathrm{D_{KL}}\left(\frac{d_{\mathrm{rk,L}}}{2n} \middle|\middle| p_\mathrm{B}\right)$, $c_2 \coloneqq \min\left\{\mathrm{D_{KL}}\left(\frac{1}{n}\left(\left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1\right) \middle|\middle| p_\mathrm{B}\right), c_1 + \frac{\mathrm{e}^{-1}}{\alpha^2}\right\}$ and $n > \frac{\mathrm{W}_{-1}(\alpha^2(c_1 - c_2))}{2(c_1 - c_2)}$. Then

$$\Pr\left\{R_+ + \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2} d_{\mathrm{rk,L}}\right\} < \Pr\left\{2R_2 + \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\right\}.$$

Let the variables and conditions given as defined and stated in the theorem. At first we separate the random variables by the law of total probability like in the proof for Theorem 4.3.3. For the lower bound of the right side of (4.9) we obtain the following:

$$\Pr\{2R_2 + \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\} = \sum_{i=0}^{n} \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) = i\right\} \underbrace{\Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}} - i}{2}\right\}}_{=1 \Leftrightarrow i \geq d_{\mathrm{rk,L}}} \tag{A.5}$$

$$= \sum_{i=0}^{d_{\mathrm{rk,L}}-1} \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) = i\right\} \underbrace{\Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}} - i}{2}\right\}}_{\geq \Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}}}{2}\right\}} +$$

$$+ \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\right\}$$

$$\geq \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\} \Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}}}{2}\right\} + \tag{A.6}$$

$$+ \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) > d_{\mathrm{rk,L}} - 1\right\}.$$

The upper bound of the other side calculates accordingly:

$$\begin{aligned}
&\Pr\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2} d_{\mathrm{rk,L}}\} \\
&\quad \leq \sum_{i=0}^{d_{\mathrm{rk,L}}-1} \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) = i\right\} \Pr\left\{R_+ \geq \frac{\rho}{2} d_{\mathrm{rk,L}} - i\right\} + \qquad \text{(A.7)} \\
&\qquad + \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) > d_{\mathrm{rk,L}} - 1\right\} \\
&\quad \leq \Pr\left\{R_+ \geq \left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1\right\} + \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) > d_{\mathrm{rk,L}} - 1\right\} \qquad \text{(A.8)}
\end{aligned}$$

It remains to be shown that

$$\Pr\left\{R_+ \geq \left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1\right\} < \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\} \Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}}}{2}\right\}, \tag{A.9}$$

which is almost the same as we have done before, except for the multiplicative term $\Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \leq d_{\mathrm{rk,L}} - 1\right\}$, which is quite hard to approximate. But we can be sure, that the term is close to 1 (cf. Section 3.1.1 (starting on page 24) and Figure 4.12 on p. 50). We use this as constant α from Lemma 4.3.2 (see p. 44), because it is $\in (0, 1]$. Since we demand $\left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1 > \frac{d_{\mathrm{rk,L}}}{2}$, we can use Lemma 4.3.2 by choosing $x_2 = \left(\frac{\rho}{2} - 1\right) d_{\mathrm{rk,L}} + 1$ and $x_1 = \frac{d_{\mathrm{rk,L}}}{2}$, which yields together with the definitions of c_1 and c_2 as given in the theorem the constraint $n > \frac{1}{2}\mathrm{W}_{-1}(\alpha^2(c_1 - c_2))/(c_1 - c_2)$.□

A.3. Proof of Theorem 5.2.5

Before the proof, we recapitulate the theorem.

Theorem 5.2.5
Let $R_1, R_2 \sim \operatorname{Bin}(n, p_{\mathrm{B}})$, let A be a random variable with the PGF given in (5.1).[2] *Further let $np_{\mathrm{B}} < \tau < n$, $\delta' > 3$, $0 < \varepsilon_\delta \leq (\delta' - 3)\tau/3$, $c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n} \middle|\middle| p_{\mathrm{B}}\right)$ and $c_2 \coloneqq \min\left\{\mathrm{D}_{\mathrm{KL}}\left(\frac{\tau+\varepsilon_\delta}{n} \middle|\middle| p_{\mathrm{B}}\right), c_1 + 4\mathrm{e}^{-1}\right\}$. It must be $n > \frac{1}{2(c_1 - c_2)}\mathrm{W}_{-1}\left(\frac{1}{4}(c_1 - c_2)\right)$ (as in Theorem 4.3.3). Further let $c \coloneqq \frac{1}{\sqrt{2n}}\mathrm{e}^{-nc_1} - 2\mathrm{e}^{-nc_2}$ and $\varepsilon_A = F_A^{-1}(1 - c)$, where $F_A^{-1} : [0, 1] \to \mathbb{R}$ is the quantile function of RV A, and $\delta = \delta' + \frac{\varepsilon_A}{\tau}$. Then*

$$\Pr\{R_1 + 2R_2 + A \geq \delta\tau\} \leq \Pr\{R_2 \geq \tau\}.$$

Let $A \coloneqq \operatorname{rk}(\Delta \boldsymbol{A}_i)$ be specified by the parameters $p_{\Delta\mathcal{N}}$, p_w and $|\mathcal{N}|$ and the PGF in (5.1). Further let $np_{\mathrm{B}} < \tau < n$, $0 < \varepsilon_\delta \leq (\delta' - 3)\tau/3$, $c_1 \coloneqq \mathrm{D}_{\mathrm{KL}}\left(\frac{\tau}{n} \middle|\middle| p_{\mathrm{B}}\right)$,

[2] i.e. specified by parameters $|\mathcal{N}|, p_w$ and $p_{\Delta\mathcal{N}}$

$c_2 \coloneqq \min\left\{\mathrm{D_{KL}}\left(\frac{\tau+\varepsilon_\delta}{n}\middle\|p_\mathrm{B}\right), c_1 + 4\mathrm{e}^{-1}\right\}$ and $n > \frac{1}{2(c_1-c_2)}\mathrm{W}_{-1}\left(\frac{1}{4}(c_1-c_2)\right)$ (as in Theorem 4.3.3). Moreover we need $c \coloneqq \frac{1}{\sqrt{2n}}\mathrm{e}^{-nc_1} - 2\mathrm{e}^{-nc_2}$ and $\varepsilon_A = F_A^{-1}(1-c)$.

Using $R_1, R_2 \sim \mathrm{Bin}\,(n,p)$, we can use the law of total probability to get

$$\Pr\left\{R_1 + 2R_2 + A \geq \delta\tau\right\} = \sum_{i=0}^{n} \Pr\left\{A = i\right\} \Pr\left\{R_1 + 2R_2 \geq \delta\tau - i\right\}. \tag{A.10}$$

We split the sum as follows

$$\begin{aligned}\Pr\left\{R_1 + 2R_2 + A \geq \delta\tau\right\} = \sum_{i=0}^{\varepsilon_A} \Pr\left\{A = i\right\} \underbrace{\Pr\left\{R_1 + 2R_2 \geq \delta\tau - i\right\}}_{\leq \Pr\{R_1+2R_2 \geq \delta\tau - \varepsilon_A\}} + \\ + \sum_{i>\varepsilon_A} \Pr\left\{A = i\right\} \underbrace{\Pr\left\{R_1 + 2R_2 \geq \delta\tau - i\right\}}_{\leq 1}\end{aligned}$$

and upper bound as already indicated:

$$\begin{aligned}\Pr\left\{R_1 + 2R_2 + A \geq \delta\tau\right\} &\leq \underbrace{\Pr\left\{R_1 + 2R_2 \geq \delta\tau - \varepsilon_A\right\}}_{=\Pr\{R_1+2R_2 \geq \delta'\tau\}} \underbrace{\sum_{i=0}^{\varepsilon_A} \Pr\left\{A = i\right\}}_{\leq 1} + \qquad (A.11) \\ &\quad + \Pr\left\{A > \varepsilon_A\right\} \\ &\leq \sum_{i=0}^{n} \Pr\left\{R_1 = i\right\} \Pr\left\{R_2 \geq \frac{\delta'\tau - i}{2}\right\} + \qquad (A.12) \\ &\quad + \Pr\left\{A > \varepsilon_A\right\}.\end{aligned}$$

For (A.12) we again use the law of total probability. In the next step we use ε_δ to part the sum once more

$$\begin{aligned}\Pr\left\{R_1 + 2R_2 + A \geq \delta\tau\right\} &= \sum_{i=0}^{\tau+\varepsilon_\delta} \Pr\left\{R_1 = i\right\} \Pr\left\{R_2 \geq \frac{\delta'\tau - i}{2}\right\} + \qquad (A.13) \\ &\quad + \sum_{i>\tau+\varepsilon_\delta} \Pr\left\{R_1 = i\right\} \underbrace{\Pr\left\{R_2 \geq \frac{\delta'\tau - i}{2}\right\}}_{\leq 1} + \\ &\quad + \Pr\left\{A > \varepsilon_A\right\} \\ &\leq \Pr\left\{R_2 \geq \frac{(\delta'-1)\tau - \varepsilon_\delta}{2}\right\} + \qquad (A.14) \\ &\quad + \Pr\left\{R_1 > \tau + \varepsilon_\delta\right\} + \Pr\left\{A > \varepsilon_A\right\}.\end{aligned}$$

Analogous to the proof of Theorem 4.3.3 we can due to $\varepsilon_\delta \leq (\delta'-3)\frac{\tau}{3}$ further upper bound the left side as follows

$$\Pr\left\{R_2 \geq \underbrace{\frac{(\delta'-1)\tau-\varepsilon_\delta}{2}}_{\geq \tau+\varepsilon_\delta}\right\} \leq \Pr\left\{R_2 \geq \tau+\varepsilon_\delta\right\}.$$

Since R_1 and R_2 are identically distributed we can write

$$\Pr\left\{R_1+2R_2+A \geq \delta\tau\right\} \leq 2\Pr\left\{R_2 \geq \tau+\varepsilon_\delta\right\} + \Pr\left\{A > \varepsilon_A\right\}.$$

From $\varepsilon_A = F_A^{-1}(1-c)$ we get

$$\begin{aligned} & \Pr\left\{A \leq \varepsilon_A\right\} \geq 1-c \\ \iff \quad & 1-\Pr\left\{A > \varepsilon_A\right\} \geq 1-c \\ \iff \quad & \Pr\left\{A > \varepsilon_A\right\} \leq c. \end{aligned}$$

By the estimations ($<$) and ($>$), which were used frequently before, we can upper bound the difference $c = \frac{1}{\sqrt{2n}}\mathrm{e}^{-nc_1} - 2\mathrm{e}^{-nc_2}$ by

$$\frac{1}{\sqrt{2n}}\mathrm{e}^{-nc_1} - 2\mathrm{e}^{-nc_2} \leq \Pr\left\{R_2 \geq \tau\right\} - 2\Pr\left\{R_2 \geq \tau+\varepsilon_\delta\right\} \tag{A.15}$$

and the claim follows. □

A.4. Proof of Theorem 5.2.7

We repeat Theorem 5.2.7 before proving it.

Theorem 5.2.7
Let $R_1, R_2 \sim \mathrm{Bin}(n, p_\mathrm{B})$, A another random variable with the PGF given in (5.1), $\alpha = \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top\right) \leq d_{\mathrm{rk,L}}-1\right\}$, $2np_\mathrm{B} < d_{\mathrm{rk,L}}$ and $\rho' > 3-\frac{2}{d_{\mathrm{rk,L}}}$, $c_1 = \mathrm{D_{KL}}\left(\frac{d_{\mathrm{rk,L}}}{2n}\middle|\middle|p_\mathrm{B}\right)$ and $c_2 = \min\left\{\mathrm{D_{KL}}\left(\frac{\left(\frac{\rho'}{2}-1\right)d_{\mathrm{rk,L}}+1}{n}\middle|\middle|p_\mathrm{B}\right), c_1+\frac{\mathrm{e}^{-1}}{\alpha^2}\right\}$ and $n > \frac{1}{2(c_1-c_2)}\mathrm{W}_{-1}\left(\alpha^2(c_1-c_2)\right)$ as in Theorem 4.4.1. Further let $c \coloneqq \frac{\alpha}{\sqrt{2n}}\mathrm{e}^{-nc_1} - \mathrm{e}^{-nc_2}$, $\varepsilon_A = F_A^{-1}(1-c)$, where $F_A^{-1}: [0,1] \to \mathbb{R}$ is the quantile function of RV A, and $\rho \geq \rho' + \frac{2\varepsilon_A}{d_{\mathrm{rk,L}}}$. Then

$$\Pr\left\{R_+ + \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top\right) + A \geq \frac{\rho}{2}d_{\mathrm{rk,L}}\right\} \leq \Pr\left\{2R_2 + \mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top\right) \geq d_{\mathrm{rk,L}}\right\}.$$

Like in the theorem before we have $R_1, R_2 \sim \mathrm{Bin}(n, p_\mathrm{B})$ and A a random variable with the PGF given in (5.1) on p. 53. We define $\alpha = \Pr\left\{\mathrm{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^\top\right) \leq d_{\mathrm{rk,L}}-1\right\}$.

Further let $2np_{\mathrm{B}} < d_{\mathrm{rk,L}}$ and $\rho' > 3 - \frac{2}{d_{\mathrm{rk,L}}}$, c_1 and c_2 as defined in the theorem and $n > \frac{1}{2(c_1-c_2)}\mathrm{W}_{-1}\left(\alpha^2(c_1-c_2)\right)$ as in Theorem 4.4.1. In addition we require $c \coloneqq \frac{\alpha}{\sqrt{2n}}\mathrm{e}^{-nc_1} - \mathrm{e}^{-nc_2}$, $\varepsilon_A = F_A^{-1}(1-c)$ and $\rho > 3 - \frac{2}{d_{\mathrm{rk,L}}}(1-\varepsilon_A)$.

As before (cf. (A.10)- (A.14)) we use the law of total probability to separate the random variables.

$$
\begin{aligned}
&\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) + A \geq \frac{\rho}{2}d_{\mathrm{rk,L}}\right\} \\
&\qquad = \sum_{i=0}^{n} \Pr\{A=i\} \Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2}d_{\mathrm{rk,L}} - i\right\} \\
&\qquad = \sum_{i=0}^{\varepsilon_A} \Pr\{A=i\} \underbrace{\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2}d_{\mathrm{rk,L}} - i\right\}}_{\leq \Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \rho d_{\mathrm{rk,L}}/2 - \varepsilon_A\right\}} \\
&\qquad\quad + \sum_{i>\varepsilon_A} \Pr\{A=i\} \underbrace{\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2}d_{\mathrm{rk,L}} - i\right\}}_{\leq 1} \\
&\qquad \leq \Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \rho d_{\mathrm{rk,L}}/2 - \varepsilon_A\right\} \underbrace{\sum_{i=0}^{\varepsilon_A} \Pr\{A=i\}}_{\leq 1} + \\
&\qquad\quad + \Pr\{A > \varepsilon_A\} \\
&\qquad \leq \Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho'}{2}d_{\mathrm{rk,L}}\right\} + \Pr\{A > \varepsilon_A\},
\end{aligned}
$$

where $\rho' \leq \rho - \frac{2\varepsilon_A}{d_{\mathrm{rk,L}}}$. We can further separate the RV similar to (A.7) and (A.8).

$$
\begin{aligned}
\Pr\left\{R_+ + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho'}{2}d_{\mathrm{rk,L}}\right\} \leq \Pr\left\{R_+ \leq \left(\frac{\rho'}{2}-1\right)d_{\mathrm{rk,L}} + 1\right\} + \\
+ \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) > d_{\mathrm{rk,L}} - 1\right\}.
\end{aligned}
$$

Analogous to the proof before we can upper bound $\Pr\{A > \varepsilon_A\}$ by c due to the definition of $\varepsilon_A = F_A^{-1}(1-c)$, where $c = \frac{\alpha}{\sqrt{2n}}\mathrm{e}^{-nc_1} - \mathrm{e}^{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{1}{\mathrm{n}}\left(\left(\frac{\rho'}{2}-1\right)\mathrm{d}_{\mathrm{rk,L}}+1\right)\middle\|\mathrm{p}_{\mathrm{B}}\right)}$, i.e.

$$
\begin{aligned}
\Pr\{A > \varepsilon_A\} &\leq \frac{\alpha}{\sqrt{2n}}\mathrm{e}^{-nc_1} - \mathrm{e}^{-n\mathrm{D}_{\mathrm{KL}}\left(\frac{1}{\mathrm{n}}\left(\left(\frac{\rho'}{2}-1\right)\mathrm{d}_{\mathrm{rk,L}}+1\right)\middle\|\mathrm{p}_{\mathrm{B}}\right)} \\
&\leq \alpha \Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}}}{2}\right\} - \Pr\left\{R_+ \leq \left(\frac{\rho'}{2}-1\right)d_{\mathrm{rk,L}} + 1\right\}
\end{aligned} \tag{A.16}
$$

(c_1 as defined above) and therefore

$$\begin{aligned}
&\Pr\left\{R_+ + A + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq \frac{\rho}{2} d_{\mathrm{rk,L}}\right\} \\
&\quad \leq \Pr\left\{R_+ \leq \left(\frac{\rho'}{2} - 1\right) d_{\mathrm{rk,L}} + 1\right\} + \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) > d_{\mathrm{rk,L}} - 1\right\} + \\
&\qquad + \Pr\left\{A > \varepsilon_A\right\} \\
&\quad \overset{(A.16)}{\leq} \Pr\left\{\operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) > d_{\mathrm{rk,L}} - 1\right\} + \alpha \Pr\left\{R_2 \geq \frac{d_{\mathrm{rk,L}}}{2}\right\} \\
&\quad \overset{(A.5)\text{-}(A.6)}{\leq} \Pr\left\{2R_2 + \operatorname{rk}\left(\boldsymbol{L}\boldsymbol{I}_{\mathcal{U}}^{\top}\right) \geq d_{\mathrm{rk,L}}\right\},
\end{aligned}$$

since the right part of the inequality stated in this theorem can be lower bounded as in the Proof of Theorem 4.4.1, where ρ' takes on the role of ρ. □

B. Implementation of the RLNC Library

The code of the RLNC channel, (de-)modulation of DLNC and other functions, which are used in the simulations, is provided in this chapter.

The RLNC Channel Model (including initialization of the channel):

```
# necessary imports
import scipy.stats  # for random variables

def init_rlnc_channel(Fq,N,n,rkA='x'):
    """
    Generate a random channel matrix A with dimensions Nxn
     input variables:
        Fq      finite field of order q
        N       first dimension of channel matrix A
        n       second dimension of channel matrix A
        rkA     defines whether channel matrix A shall
                have full rank (option 'n') or an
                arbitrary rank (option 'x' or anything else
                than 'n')
     output:
        A   channel matrix Nxn-matrix, (uniformly) randomly
            chosen from all (or regular) matrices
    """

    MS = MatrixSpace(Fq,N,n);

    if rkA == 'n':
        A = matrix.random_echelonizable(MS,rank=min(n,N));
    else:
        A = MS.random_element();

    return A

def rlnc_channel(A,X,p_B,numNodes,p_L,p_W,rkDelta=0,debug=False):
    """
    Realisation of an RLNC Channel
     input:
        A           is the channel matrix of dimension N x n
```

```
        X           is the sending matrix of dimension n x M
        p_B         is the probability for the Bin(n,p_B)
                    distributed error matrix B
        numNodes    number of nodes in the network,
                    numNodes >> n must hold
        p_L         probability for number of leaving nodes
                    L~Bin(numNodes,p_L)
        p_W         probability for node weight W~Bin(numNodes,p_W)
        rkDelta     decides whether to change channel matrix
                    rkDelta = 0:    no change

     output:
        Y = A*X + B is the receive matrix
        A           is the perhaps changed channel matrix

      components of Y:
        B is an nx(n+m)-matrix, s.t.

                rank(B) = t ~ Bin(n,p_B),

        DeltaA is an Nxn-matrix, s.t.

                rank(DeltaA) <= sum_{i=1}^L W(v_i),

        where L~Bin(numNodes,p_L), W~Bin(numNodes,p_W)

    Source for the calculation of the channel
    deviation (DeltaA): Puchinger, Cyran:
        "Error Correction for Differential Linear Network
         Coding in Slowly-Varying Networks"
    """

    # check dimensions
    if A.dimensions()[1] != X.dimensions()[0]:
        raise ValueError("dimensions of A and X don't fit.")

    # Definitions
    N  = A.dimensions()[0];
    n  = X.dimensions()[0];
    M  = X.dimensions()[1];
    Fq = X.parent().base();

    # change channel matrix if desired
    if rkDelta == 0:
        DeltaA = zero_matrix(Fq,N,n); # zero matrix
    else: # rk(DeltaA) <= sum_i=1^L W(v_i), see source above
        # define random variable L:
        rvL = scipy.stats.binom(numNodes,p_L);
        l   = rvL.rvs(1)[0];# generate random number
        # define random variable W:
```

```
        rvW = scipy.stats.binom(numNodes,p_W);
        rk  = sum(rvW.rvs(1));# rk(DeltaA)
        if rk > min(n,N):
            rk = min(n,N);
        if debug:
            print 'rk(DeltaA) = ', rk
        DeltaA = random_matrix(Fq,N,n,algorithm=\
                               'echelonizable',rank=rk);
    A = A + DeltaA;

    # choose error rank t randomly from Bin(n,p_B)
    # define random variable for tau:
    rvt = scipy.stats.binom(n,p_B);
    t   = rvt.rvs(1)[0];# generate random number

    # error matrix B with rank t
    B = random_matrix(Fq,N,M,algorithm='echelonizable',rank=t);

    return (A*X + B, A)
```

The Lifting Construction:

```
def lifting(S):
    """
    Perform lifting construction for an information matrix S
    Reference: Silva, Kschischang, Kötter:
        "A Rank-Metric Approach to Error Control in
        Random Network Coding"
    """

    n = S.dimensions()[0];
    Fq = S.parent().base();
    return block_matrix(1,2,[identity_matrix(Fq,n),S])
```

The implementations of DLNC modulation, demodulation and necessary functions:

```
def dlnc_modulation(X_imin1,S_i):
    """
    Modulation of differentially encoded matrix X,
    i.e. X_i = [X_{i-1}]_n * S_i
    S_i is the information as in Seidl's Paper:
        "A Differential Encoding Approach to
         Random Linear Network Coding"
    input:
        X_imin1     transmit matrix from previous step,
                    size nx(n+m)
        S_i         information matrix, size nx(n+m)

    output:
        X_i         current transmit matrix
    """

    # first dimension of X_i
    n = X_imin1.dimensions()[0];

    # check dimensions
    if n > X_imin1.dimensions()[1]:
        raise ValueError("X_imin1 must be a fat or square matrix.")
   ;
    if n != S_i.dimensions()[0]:
        raise ValueError("Dimensions must fit.");

    return X_imin1[:,0:n]*S_i

def dlnc_mod(X_imin1,S_i):
    return dlnc_modulation(X_imin1,S_i);

def dlnc_demodulation(Y_i,Y_imin1):
    """
    Demodulation of differentially encoded matrix X_i,
    i.e. X_i = [X_i]_n * S_i
    S_i is the information as in Seidl's Paper:
    "A Differential Encoding Approach to
     Random Linear Network Coding"
    the demodulation works as follows:

            S_hat = (Y_imin1)_[n]^+ * Y_i

    where (Y_imin1)_[n]^+ is the pseudoinverse
    of the first n columns of Y_imin1, fulfilling

            (Y_imin1)_[n]^+ * (Y_imin1)_[n] = I_n + L*I_U^T,

    where I_U^T*L = -I_|U|x|U| (cf. function weakpseudoinverse() )
    input:
```

```
49          Y_i             received matrix in step i
                            with dimensions nx(n+m)
51          Y_imin1         received matrix of previous step i-1
                            with dimensions nx(n+m)
53
    output:
55          S_hat           demodulated matrix, i.e. estimate for
                            information matrix (coded or uncoded)
57      """
59      # first dimension of Y_i
        n = Y_i.dimensions()[0];
61
        # Remark: Y_imin1_inv's dimensions are chosen s.t.
63      #         they fit Y_i's
        Y_imin1_inv = weakpseudoinverse(Y_imin1[0:n,0:n]);
65      return Y_imin1_inv*Y_i

67  def dlnc_demod(Y_i,Y_imin1):
        return dlnc_demodulation(Y_i,Y_imin1);
69
    def weakpseudoinverse(Y,debug=False):
71      """
        Compute (weak) pseudoinverse of Y according to
73      Seidl et. al.:
            "A Differential Encoding Approach to
75           Random Linear Network Coding"
        and Silva et. al.:
77          "A Rank-Metric Approach to Error Control
             in Random Network Coding"
79
        input:
81          Y       square matrix of dimensions nxn
        output:
83          Y_pinv  square matrix of dimensions nxn,
                    satisfying
85
                Y_pinv*Y = I + L*I_U^T
87
        where I is an nxn-identity matrix, U is a subset of
89      the columns of I and L satisfies

91          I_U^T * L = -I_|U|x|U|

93      Remark: Currently this only works for matrices over
                GF(2^l) for any l>1
95      """

97      # Definitions and checks
        Fq      = Y.parent().base();
```

```
99      if Fq.characteristic() != 2 or Fq.cardinality() == 2:
            raise ValueError("Matrix must be defined over Field of
    characteristic 2 but not GF(2).");
101
        n1,n2   = Y.dimensions();
103     if n1 != n2:
            raise ValueError("Input matrix must be square.")
105
        # calculate inverse
107     Y_pinv = Y.inverse();

109     # if Y is invertible, the pseudoinverse is unambigious,
        # i.e. the same as the actual inverse
111     if Y.is_invertible():
            return Y_pinv
113
        #check if inverse is suitable
115     I_L     = Y_pinv*Y;
        W       = I_L.rref();
117     U,U_c   = find_leading_positions(W,n1);
        W       = W[0:W.rank()]; # remove zero rows
119     I_U     = identity_matrix(Fq,n1)[:,U];
        I_U_c   = identity_matrix(Fq,n1)[:,U_c];
121     L       = -I_U + I_U_c * W * I_U;
        I_UxU   = identity_matrix(Fq,len(U));
123     if I_L != identity_matrix(Fq,n1) + L*I_U.transpose() \
        or I_U.transpose()*L != -I_UxU:
125         if debug:
                print "inverse does not fulfill conditions --> swap
    rows"
127         for i in U_c: # go through diagonal of Y_p*Y = I_L
                if I_L[i,i] != 1:
129                 for j in xrange(n1): # go through column
                        if I_L[j,i] == 1:
131                         Y_pinv.swap_rows(i,j);
                            I_L = Y_pinv*Y;
133                         break;

135     if Y_pinv.rank() < n1 \
        or I_L != identity_matrix(Fq,n1) + L*I_U.transpose() \
137     or I_U.transpose()*L != -I_UxU:
            raise ValueError(\
139           "Could not find pseudoinverse with desired properties.")

141     return Y_pinv

143 def find_leading_positions(A,n):
        """
145     Find the positions of the leading coefficients of the
        reduced row echelon form of matrix A
```

```
    input:
        A       matrix of dimensions  s x l
                (A must be a fat or square matrix)
        n       defines the set {1,...,n}
    output:
        U_c     set consisting of the positions of the
                leading coefficients in {1,...,n}
        U       set U = {1,...,n}\U_c
    Remark: U ist returned first
    """

    # sanity checks
    n1,n2 = A.dimensions()
    if n1 > n2:
        raise ValueError("A must be a fat or square matrix.");
    if n > n2:
        raise ValueError(\
          "n must be smaller or equal the second dimension of A.");

    A = A.rref();
    U_c = [];
    for r in xrange(min(n,A.rank())): # go through rows
        for c in xrange(r,n):  # go through row r
            if A[r,c] == 1:
                U_c.append(c); # found leading position
                break;

    U   = [x for x in xrange(n) if x not in U_c];
    return U,U_c;
```

Functions for the check of decodability:

```
def is_decodable(mode,*args):
    """
    Function that checks the decoding condition for an
    encoding scheme specified by "mode"
    options for "mode":
     *"DLNC" :  check if the rank of the effective error
                matrix of a differentially encoded matrix
                is small enough, i.e. if

                        2*rk(S_hat - S) <= d-1

        requires the following input parameters:
        S       information matrix, dimensions: nx(n+m)
        Y_i     current received matrix, dimensions: nx(n+m)
        Y_imin1 current error matrix, dimensions: nx(n+m)
        d       minimum distance of the used rank-metric code

     *"Lifting": checks if one can guarantee a  decoding
```

```
                success for the received matrix corresponding
                to a lifted codeword, i.e. if

            _     _
           |       |
           |   X   |
       2rk |       | - rk ( X ) - rk( Y ) <= d-1
           |   Y   |
           |_     _|

       source: Silva,Kschischang, Kötter:
       "A Rank-Metric Approach to  Error Control in
        Random Network Coding"
    input:
       X   transmit matrix, dimensions: nx(n+m)
       Y   received matrix, dimensions: N x (n+m)
       d   minimum distance of the applied rank-metric code
    return: boolean True if decodable, False if not
    """
    if mode == "DLNC":
        return is_dlncdecodable(*args);
    if mode == "Lifting":
        return is_liftdecodable(*args);
    else:
        raise AttributeError("Specify mode.");

def is_dlncdecodable(*args):
    """
    Function validates decoding condition for DLNC to
    check if the rank of the effective error matrix of
    a differentially encoded matrix is small enough,
    i.e. if

                        2*rk(S_hat-S) <= d-1

    input:
       S       information matrix, dimensions: nx(n+m)
       Y_i     current received matrix, dimensions: nx(n+m)
       Y_imin1 current error matrix, dimensions: nx(n+m)
       d       minimum distance of the rank-metric code
    return: boolean True if decodable, False if not
    """

    # check number of input arguments
    numargs = 4;
    if len(args) != numargs:
        raise ValueError("Number of arguments must be " +\
                         str(numargs)+ " not " +\
                         str(len(args)))

    # extract parameters
```

```
    S       = args[0];
    Y_i     = args[1];
    Y_imin1 = args[2];
    d       = args[3];

    # check parameter dimensions e.g. d must be scalar
    if  isinstance(d, (list,str,unicode)):
        raise TypeError("d should be scalar.");

    if S.dimensions()[0] != Y_i.dimensions()[0]:
        raise ValueError(\
            "S and Y must have the same first dimension");

    # demodulate DLNC
    S_hat = dlnc_demod(Y_i,Y_imin1);

    # check if 2rk(E_i) = 2rk( S_hat - S) <= d-1
    if 2*(S_hat-S).rank() <= d-1:
        return True;
    else:
        return False;

def is_liftdecodable(*args):
    """
    Function validates decoding condition for Lifting
    construction, it checks if one can guarantee a
    decoding success for the received matrix corresponding
    to a lifted codeword, i.e. if

            _     _
            |     |
            |  X  |
        2rk |     | - rk ( X ) - rk( Y ) <= d-1
            |  Y  |
            |_    _|

        source: Silva,Kschischang, Kötter:
        "A Rank-Metric Approach to  Error Control in
         Random Network Coding"
     input:
        X   transmit matrix, dimensions: nx(n+m)
        Y   received matrix, dimensions: N x (n+m)
        d   minimum distance of the applied rank-metric code
     return: boolean True if decodable, False if not
    """

    # check number of input arguments
    numargs = 3;
    if len(args) != numargs:
        raise ValueError("Number of arguments must be " +\
                         str(numargs));
```

```
119
    # extract parameters
121 X = args[0];
    Y = args[1];
123 d = args[2];

125 # check parameter dimensions e.g. d must be scalar
    if  isinstance(d, (list,str,unicode)):
127     raise TypeError("d should be scalar.");

129 if X.dimensions()[1] != Y.dimensions()[1]:
        raise ValueError("Dimensions do not fit.");
131
    if 2*block_matrix([[X],[Y]]).rank()-X.rank()-Y.rank()<d:
133     return True;
    else:
135     return False;
```

Bibliography

[ACLY00] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46:1204 – 1216, July 2000.

[Ash67] Robert. B. Ash. *Information Theory*. Number 19 in Interscience Tracts in Pure and Applied Mathematics. Interscience Publishers, 1967.

[Bos13] Martin Bossert. *Kanalcodierung*. Oldenbourg Verlag München, 3rd edition, 2013.

[CGH+96] Robert M. Corless, Gaston H. Gonnet, D. E. G. Hare, David J. Jeffrey, and Donald E. Knuth. On the Lambert W Function. *Advances in Computational Mathematics*, 2:329–359, 1996.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

[CX02] K. P. Choi and Aihua Xia. Approximating the number of successes in independent trials: Binomial versus poisson. *Ann. Appl. Probab.*, 12(4):1139–1148, November 2002.

[Cyr17] Michael Cyran. *Channel Coding and Precoding for Linear Network Coding*. doctoralthesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2017.

[Del78] Philippe Delsarte. Bilinear Forms over a Finite Field, with Applications to Coding Theory. *Journal of Combinatorial Theory, Series A*, 25(3):226 – 241, 1978.

[Eul83] Leonard Euler. De serie lambertina plurimisque eius insignibus proprietatibus. *Acta Acad. Scient. Petropol.*, 2:29–51, 1783. Reprinted in Euler, L. Opera Omnia, Series Prima, Vol. 6: Commentationes Algebraicae. Leipzig, Germany: Teubner, pp. 350–369, 1921.

[FS07] Christina Fragouli and Emina Soljanin. Network Coding Fundamentals. In *Foundations and Trends in Networking*, 2007.

[Gab85] Ernst M. Gabidulin. Theory of Codes with Maximum Rank Distance. *Probl. Peredachi Inf.*, 21(1):3–16, 1985.

[HL08] Tracey Ho and Desmond Lun. *Network Coding: An Introduction*. Cambridge University Press, 2008.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[KK08] Ralf Kötter and Frank R. Kschischang. Coding for Errors and Erasures in Random Network Coding. *IEEE Transactions on Information Theory*, 54(8):3579–3591, August 2008.

[KL51] Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *Ann. Math. Statist.*, 22(1):79–86, March 1951.

[KRH+06] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. XORs in The Air: Practical Wireless Network Coding. *SIGCOMM*, 2006.

[KS98] Roelof Koekoek and René F. Swarttouw. The askey-scheme of hypergeometric orthogonal polynomials and its q-analogue, 1998.

[Lam58] Johann Heinrich Lambert. Observationes Variea in Mathesin Puram. *Acta Helvetica, physico-mathematico-anatomico-botanico-medica*, pages 128–168, 1758.

[LN83] Rudolf Lidl and Harald Niederreiter. *Finite Fields (Encyclopedia of Mathematics)*, volume 20. Addison-Wesley, 1983.

[Mey00] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. 2000.

[Ore33] Oystein Ore. On a special class of polynomials. *Trans. Amer. Math. Soc.*, 35:559–58, 1933.

[Ove07] Raphael Overbeck. *Public Key Cryptography based on Coding Theory*. PhD thesis, Technische Universität Darmstadt, 2007.

[PCF+15] Sven Puchinger, Michael Cyran, Robert F. H. Fischer, Martin Bossert, and Johannes B. Huber. Error Correction for Differential Linear Network Coding in Slowly-Varying Networks. In *SCC 2015; 10th International ITG Conference on Systems, Communications and Coding*, pages 1–6, February 2015.

[Ros] Johan Rosenkilde. *Codinglib*. `https://bitbucket.org/jsrn/codinglib`.

[Rot91] Ron M. Roth. Maximum-Rank Array Codes and the Their Application to Crisscross Error Correction. 37(2):328 – 336, April 1991.

[S+18] William A. Stein et al. *SageMath Software Version 7.6*. The Sage Developers, 2018. `http://www.sagemath.org`.

[SCFH13] Mathis Seidl, Michael Cyran, Robert F. H. Fischer, and Johannes B. Huber. A Differential Encoding Approach to Random Linear Network Coding. In *SCC 2013; 9th International ITG Conference on Systems, Communication and Coding*, pages 1–6, January 2013.

[Sha48] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell system technical journal*, 27:379–423, July 1948.

[SKK08] Danilo Silva, Frank R. Kschischang, and Ralf Kötter. A Rank-Metric Approach to Error Control in Random Network Coding. *(published in) IEEE Transactions of Information Theory*, 54, September 2008.

[Spo14] Evgeny Spodarev. Wahrscheinlichkeitsrechnung. Vorlesungsskript Uni Ulm, 2014.

[vLW01] Jacobus H. van Lint and Richard M. Wilson. *A Course in Combinatorics*. Cambridge University Press, November 2001.

[Wac13] Antonia Wachter-Zeh. *Decoding of Block and Convolutional Codes in Rank Metric*. doctoralthesis, Ulm University, March 2013.