On Linear Transmission Systems

Kapetanovic, Dzevdan

2012

Thesis

On Linear Transmission Systems

Dževdan Kapetanović

Lund 2012

Department of Electrical and Information Technology
Lund University
Box 118, SE-221 00 LUND
SWEDEN

This thesis is set in Computer Modern 10pt
with the LaTeX Documentation System

Series of licentiate and doctoral theses
No. 43
ISSN 1654-790X

Printed in Sweden by *Tryckeriet i E-huset*, Lund.
June 2012.

To my mother...

Sammanfattning

Avhandlingen är uppdelad i två delar. Del I analyserar kapacitet för linjära modulationssystem med en bärvåg. Kapaciteten är en övre gräns på antalet bitar som kan överföras under en användning av kommunikationskanalen, och uppnås utav Gaussiska symboler. Den beror på den underliggande pulsen i ett linjärt modulationssystem och också signaleringshastigheten, d.v.s., hastigheten som de Gaussiska symbolerna skickas med. Målet i Del I är att studera pulsens och signaleringshastighetens påverkan på kapaciteten.

Del II i avhandlingen ägnas åt Multipel Antenna System (MIMO), och mer specifikt åt linjära förkodare för MIMO system. Linjär förkodning är ett praktiskt sätt att förbättra prestandan av ett MIMO system och har studerats omfattande under de fyra senaste decennierna. I praktiska applikationer, så är symbolerna som skickas tagna från ett diskret alfabet, så som QAM, och målet är att hitta optimala linjära förkodare för ett visst prestandamått av MIMO kanalen. Designproblemet beror på prestandamåttet och även mottagarstrukturen. Svårigheten med att hitta optimala förkodare beror på problemets diskreta natur, och hittills har suboptimala lösningar mestadels föreslagits. Problemet är väl undersökt i fallet med linjära mottagare, och optimala förkodare har hittats för många olika prestandamått i detta fallet. Dock under användning av en optimal mottagare (ML mottagare), har det hittills bara föreslagits suboptimala förkodare konstruktioner. Del II i avhandlingen börjar med att föreslå nya, lågkomplexitets suboptimala förkodare, som resulterar i en låg bitfelssannolikhet (BER) hos mottagaren. Därefter utvecklas en iterativ optimeringsmetod, vilken producerar förkodare som förbättrar de hittills bästa i litteraturen. De erhållna förkodarna uppvisar en viss struktur, som därefter analyseras och visas vara optimal för stora symbol alfabet. Dessa resultat visas också vara tillämpbara för små, praktiska symbol alfabet, och ger upphov till nya sätt att konstruera förkodare med utmärkt prestanda för ML mottagare.

Abstract

This thesis is divided into two parts. Part I analyzes the information rate of single antenna, single carrier linear modulation systems. The information rate of a system is the maximum number of bits that can be transmitted during a channel usage, and is achieved by Gaussian symbols. It depends on the underlying pulse shape in a linear modulated signal and also the signaling rate, the rate at which the Gaussian symbols are transmitted. The object in Part I is to study the impact of both the signaling rate and the pulse shape on the information rate.

Part II of the thesis is devoted to multiple antenna systems (MIMO), and more specifically to linear precoders for MIMO channels. Linear precoding is a practical scheme for improving the performance of a MIMO system, and has been studied intensively during the last four decades. In practical applications, the symbols to be transmitted are taken from a discrete alphabet, such as quadrature amplitude modulation (QAM), and it is of interest to find the optimal linear precoder for a certain performance measure of the MIMO channel. The design problem depends on the particular performance measure and the receiver structure. The main difficulty in finding the optimal precoders is the discrete nature of the problem, and mostly suboptimal solutions are proposed. The problem has been well investigated when linear receivers are employed, for which optimal precoders were found for many different performance measures. However, in the case of the optimal maximum likelihood (ML) receiver, only suboptimal constructions have been possible so far. Part II starts by proposing new novel, low complexity, suboptimal precoders, which provide a low bit error rate (BER) at the receiver. Later, an iterative optimization method is developed, which produces precoders improving upon the best known ones in the literature. The resulting precoders turn out to exhibit a certain structure, which is then analyzed and proved to be optimal for large alphabets.

Preface

Part I of the thesis is based on the following papers.

[1] D. Kapetanović and F. Rusek, "The Effect of Signaling Rate on Capacity for Linear Transmission Systems," *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 421–428, February 2012.

[2] D. Kapetanović, F. Rusek, J. B. Anderson, "The Effect of Symbol Rate on Constrained Capacity for Linear Modulation," In *Proc. IEEE International Symposium on Information Theory*, Toronto, Canada, July 2008.

Part II of the thesis is based on the following papers.

[3] D. Kapetanović, F. Rusek, T. Abrudan, V. Koivunen, "Construction of Minimum Euclidean Distance MIMO Precoders and their Lattice Classifications," accepted for publication in *IEEE Transactions on Signal Processing*.

[4] D. Kapetanović, H. V. Cheng, W. H. Mow, F. Rusek, "Optimal Two-Dimensional Lattices for Precoding of Linear Channels," under review in *IEEE Transactions on Wireless Communication*.

[5] D. Kapetanović, H. V. Cheng, W. H. Mow, F. Rusek, "A Lattice-Theoretic Characterization of Optimal Minimum-Distance Linear Precoders," submitted to *IEEE Transcations on Information Theory*.

[6] D. Kapetanović, H. V. Cheng, W. H. Mow, F. Rusek, "Optimal lattices for MIMO precoding," In *Proc. IEEE International Symposium on Information Theory*, St. Petersburg, Russia, July 2011.

[7] D. Kapetanović and F. Rusek, "A Comparison Between Unitary and Non-Unitary Precoder Design for MIMO Channels with MMSE Detection and Limited Feedback," In *Proc. IEEE Global Communications Conference 2009 (GLOBECOM)*, Miami, FL, December 2010.

[8] F. Rusek and D. Kapetanović, "Design of close to optimal Euclidean distance MIMO-precoders," In *Proc. IEEE International Symposium on Information Theory*, Seoul, Korea, June 2009.

Morover, following work has also been published by the author

[9] F. Rusek and D. Kapetanović, "Optimal time–frequency occupancy of finite packet OFDM," in *Proc. IEEE Personal, Indoor and Mobile Radio Conference 2007 (PIMRC)*, Sept. 2007.

[10] F. Rusek, A. Prlja, D. Kapetanović, "Faster-than-Nyquist signaling based on short finite pulses," Radiovetenskaplig konferens 2008 (RVK'08), Växjö, Sweden.

[11] D. Kapetanović and F. Rusek, "On Precoder Design under Maximum-Likelihood Detection for Quasi-Stationary MIMO Channels," In *Proc. International Conference on Communications (ICC)*, Cape Town, SA, May 2010.

[12] D. Kapetanović and F. Rusek, "Linear precoders for parallell Gaussian channels with low decoding complexity," In *Proc. IEEE Vehicular Technology Conference-Fall*, San Francisco, Sept. 2011.

Acknowledgements

First and foremost, I would like to thank my supervisors for making this thesis possible. Prof. John B. Anderson is my main supervisor, and I am grateful and indebted to him for accepting me as one of his Ph. D. students. His calm approach to things gave me perspective, and all the advice, support and importantly freedom in research that he gave to me made it possible to pursue my own research interests. The time I started as a Ph. D. student, Dr. Fredrik Rusek was finishing his Ph. D., and he stayed at the department as a Researcher and is currently an Associate Professor. John realized that Fredrik and I shared common research interests, and thus mainly handed over my research supervision to Fredrik, who then officially became my second supervisor. This is simply what was decided upon, and it turned out to be a perfect match! Fredrik's ability to engage into research discussions with me and his enthusiasm for research stimulated my interest in communication theory, which made this thesis possible. Beside inducing me with research energy, without his exceptional brilliance, ideas and deep insights, I would fumble in the dark for a very long time during my years as a Ph. D. Student. He is truly a researcher of world class, and is a key person that made all this research possible. Therefore, I will always be grateful to Fredrik for all his efforts that finally directed me onto a good research path!

Beside my main supervisors in Lund, I would also like to thank a lot my collaborators in Hong Kong, Prof. Wai Ho Mow and Hei Victor Cheng, whom I got the opportunity to visit in Hong Kong for 4 months. Without Prof. Mow's key insights into lattice theory, a big portion of this thesis would have not been realized. I worked intensively with his student Hei Victor, and this time I will always rememeber, since the most fruitful research resulted from our collaboration. Therefore, a big thanks to my Hong Kong collaborators!

Further fruitful collaboration was also done with Dr. Traian Emanuel Abrudan and Prof. Visa Koivunen, and I would like to thank Traian for being very helpful and providing me with nice research ideas.

Although research is fun, a balanced life is very important for me. Thus,

I would like to thank all my friends that I spent time with after the working hours. Moreover, I would also like to thank all the Ph. D. students and the staff at the EIT department, which gave me a very nice working environment. Many names should be listed, but I choose not to do so, since these people feel the gratitude I have towards them.

Finally, the person who I dedicate this thesis to, the most important person in my life: my mother, Fatima Kapetanović. Without her constant support, caring and love, I would have never reached this stage of my life. Truly, my accomplishments are very much a result of all the efforts she made during my life to guide me onto the right path. Thus, with all love and thankfulness, I end these acknowledgments with a thanks to my mother.

Dževdan Kapetanović

Contents

Part I

Analysis of Information Rate for Linear Modulation Systems

Chapter 1

The Impact of Signaling Rate on Information Rate for Linear Modulation Systems

In the first part of this thesis, we study single carrier linear modulation systems. More precisely, we are interested in the achievable Shannon rates for these systems. It turns out that these Shannon rates heavily depend on the signaling rate in the linear modulation, and it is of interest in Part I to study this dependence. At the time being, Orthogonal Frequency Division Multiplexing (OFDM) systems are dominant in emerging wireless standards, but important applications of single carrier systems exist [1, 2, 3, 4, 5, 6, 7]. The most prominent contemporary applications of linear modulation are optical communications and the uplink of the LTE standard [8][1]. This chapter starts by giving a brief introduction to single carrier linear modulation systems in Section 1.1. Section 1.2 derives the Shannon information rate[2] for the studied system. A connection between the signaling rate and the information rate is noted, and thereafter an analysis is conducted to investigate this connection. This analysis is performed in Section 1.4. New results arise from this analysis, that provide necessary and sufficient pulse conditions in order to have an

[1] In LTE, the chosen modulation is single carrier frequency division multiple access (SC-FDMA), which is a multicarrier modulation but with single carrier properties.

[2] Information rate in the classical Shannon sense, i.e., the maximum number of bits/channel use that can be transmitted for a given modulation scheme.

increasing information rate with respect to the signaling rate.

1.1 Mathematical Model of Single Carrier Linear Modulation Systems

This section starts by formulating a mathematical model for single carrier linear modulation systems in Section 1.1.1. Section 1.1.2 describes the channel over which communiation occurs, while Section 1.1.3 formulates the receiver which gives rise to an equivalent discrete time model of the communication system.

1.1.1 Single Carrier Linear Modulation

A general, single carrier time-varying signal $x_c(t)$ transmitted from one antenna device can be expressed as

$$x_c(t) = \sqrt{2}\,\mathrm{Re}\{x(t)e^{i2\pi f_c t}\}, \tag{1.1}$$

where $\mathrm{Re}\{\cdot\}$ denotes the real-part of a complex number, f_c is the *carrier frequency* and $x(t)$ is the complex-valued *baseband signal* that carries all the information content to be conveyed to a receiving device. The baseband signal has its frequency support concentrated around 0, and it is further assumed to be *bandlimited* to W positive Hz, i.e., $X(f) = 0$ for $|f| > W$, where $X(f) = \mathcal{F}\{x(t)\}$ is the Fourier transform of $x(t)$ and W the *bandwidth* of $x(t)$. It is assumed that $W \ll f_c$, so that no frequency overlap occurs in the transmitted signal. Upon transmission, its spectrum is shifted to the carrier frequency f_c, in order to accomodate frequency requirements on the system. Several methods to construct the baseband signal $x(t)$ exist, and we focus on the simple and practical *linear modulation*, where $x(t)$ is expressed as

$$x(t) = x_{\boldsymbol{A}}(t,T) \stackrel{\triangle}{=} \sqrt{P_0 T} \sum_{j=-\infty}^{\infty} a_j h(t - jT). \tag{1.2}$$

In (1.2), $\boldsymbol{a} = \{\ldots, a_{-1}, a_0, a_1, \ldots\}$ is the sequence of complex-valued information bearing data symbols and $h(t)$ is a real-valued modulation pulse. The sequence $\boldsymbol{a}$ is a realization of a sequence of random variables $\boldsymbol{A} = \{\ldots, A_{-1}, A_0, A_1, \ldots\}$; thus, $x(t)$ is a random process. P_0 is the average transmitted power of $x_{\boldsymbol{A}}(t,T)$

$$P_0 \stackrel{\triangle}{=} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left\{\int_0^T |x_{\boldsymbol{A}}(t,T)|^2\,\mathrm{d}t\right\}. \tag{1.3}$$

Here and throughout the thesis, $\mathbb{E}\{\cdot\}$ denotes the expectation operator. As will be discussed shortly, (1.3) imposes constraints on $h(t)$ and the random vector $\boldsymbol{A}$. The bandlimitation of $x_{\boldsymbol{A}}(t,T)$ is incurred by bandlimitting $h(t)$ to W Hz. Further, we assume that $h(t)$ is unit energy. Hence

$$\begin{aligned} \int_{-\infty}^{\infty} |h(t)|^2 \,\mathrm{d}t &= 1 \\ H(f) &= 0, \quad |f| > W. \end{aligned} \tag{1.4}$$

The autocorrelation of $x_{\boldsymbol{A}}(t,T)$ is defined as

$$\begin{aligned} \phi_{x_{\boldsymbol{A}}}(\tau+t,t) &\stackrel{\triangle}{=} \mathbb{E}\{x_{\boldsymbol{A}}(t+\tau,T)x^*_{\boldsymbol{A}}(t,T)\} \\ &= P_0 T \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(t+\tau-jT)h^*(t-kT)\mathbb{E}\{a_j a_k^*\}, \end{aligned} \tag{1.5}$$

where $*$ is complex conjugation. Since $x_{\boldsymbol{A}}(t,T)$ is a wide-sense cyclostationary process with period T, its time-average autocorrelation function is

$$\begin{aligned} \phi_{x_{\boldsymbol{A}}}(\tau) &\stackrel{\triangle}{=} \frac{1}{T}\int_0^T \phi_{x_{\boldsymbol{A}}}(\tau+t,t)\,\mathrm{d}t \\ &= P_0 \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \mathbb{E}\{a_j a_k^*\} \int_0^T h(t+\tau-jT)h^*(t-kT)\,\mathrm{d}t. \end{aligned} \tag{1.6}$$

By making the variable substitution $k = j+p$, and defining $R_p \stackrel{\triangle}{=} \mathbb{E}\{a_j a_{j+p}^*\}$ (the correlation among symbols $\{a_j\}$ only depends on their relative position, since we assume that $\boldsymbol{A}$ is a stationary process), we arrive at

$$\begin{aligned} \phi_{x_{\boldsymbol{A}}}(\tau) &= P_0 \sum_{p=-\infty}^{\infty} R_p \sum_{j=-\infty}^{\infty} \int_{-jT}^{(1-j)T} h(t+\tau)h^*(t-pT)\,\mathrm{d}t \\ &= P_0 \sum_{p=-\infty}^{\infty} R_p \int_{-\infty}^{\infty} h(t+pT+\tau)h^*(t)\,\mathrm{d}t. \end{aligned} \tag{1.7}$$

By taking the Fourier transform of (1.7), we obtain the power spectral density (PSD) of $x_{\boldsymbol{A}}(t,T)$

$$\begin{aligned} S_{X_{\boldsymbol{A}}}(f,T) &\triangleq \mathcal{F}\{\phi_{x_{\boldsymbol{A}}}(\tau)\} \\ &= P_0|H(f)|^2 \sum_{p=-\infty}^{\infty} R_p e^{-i2\pi fpT} \\ &= P_0|H(f)|^2 S_{\boldsymbol{a}}(f,T), \quad |f|<W, \end{aligned} \tag{1.8}$$

where

$$S_{\boldsymbol{A}}(f,T) \triangleq \sum_{p=-\infty}^{\infty} R_p e^{-i2\pi fpT}. \tag{1.9}$$

$S_{\boldsymbol{A}}(f,T)$ is periodic with period $f = 1/T$, and from Parseval's identity, it follows that the average power of $x_{\boldsymbol{A}}(t,T)$ is

$$\int_{-W}^{W} S_{X_{\boldsymbol{A}}}(f,T)\,\mathrm{d}f = P_0 \int_{-W}^{W} S_{\boldsymbol{A}}(f,T)|H(f)|^2\,\mathrm{d}f.$$

Note that $|H(f)|^2$ is symmetric around $f = 0$ since $h(t)$ is a real-valued pulse. Thus, in order for P_0 to be the average power, the following identity must hold

$$\int_{-W}^{W} S_{\boldsymbol{A}}(f,T)|H(f)|^2\,\mathrm{d}f = 1. \tag{1.10}$$

From now on, $S_X(f)$ will denote the PSD of a signal $x(t)$.

1.1.2 Frequency selective and non-selective channels

If the frequency modulated signal $x_c(t)$ in (1.1) is subject to a multipath environment, represented by a real-valued impulse response $g_c(t)$, the received signal $r_c(t)$ becomes [9]

$$\begin{aligned} r_c(t) &= \int_{-\infty}^{\infty} g_c(\tau)x_c(t-\tau)\,\mathrm{d}\tau + n_c(t) \\ &= \mathrm{Re}\left\{\left[\int_{-\infty}^{\infty} g_c(\tau)e^{-i2\pi f_c\tau}x_{\boldsymbol{A}}(t-\tau,T)\,\mathrm{d}\tau\right]e^{i2\pi f_c t}\right\} + n_c(t). \end{aligned} \tag{1.11}$$

In (1.11), $n_c(t)$ is additive white Gaussian noise (AWGN) with zero-mean $\mathbb{E}\{n_c(t)\} = 0$ and autocorrelation $\mathbb{E}\{n_c(t)n_c(t+\tau)\} = N_0\delta(\tau)$, where $\delta(\tau)$ is the Kroenecker delta function. We can write $n_c(t) = \sqrt{2}\,\mathrm{Re}\{n(t)e^{i2\pi f_c t}\}$, where

$n(t)$ is a complex-valued AWGN with mean $\mathbb{E}\{n(t)\} = 0$ and autocorrelation $\mathbb{E}\{n(t)n^*(t+\tau)\} = N_0\delta(\tau)$. Upon defining

$$g(\tau) \stackrel{\triangle}{=} g_c(\tau)e^{-i2\pi f_c\tau}, \tag{1.12}$$

we see that the integral in (1.11) represents the convolution of $x_{\boldsymbol{A}}(t,T)$ with a complex-valued baseband channel impulse response $g(\tau)$. By inserting the expression for $x_{\boldsymbol{A}}(t,T)$ in (1.2) into (1.11), the complex baseband model of (1.11) becomes

$$\begin{aligned} r(t) &= v_{\boldsymbol{A}}(t,T) + n(t) \\ &= g(t) \star x_{\boldsymbol{A}}(t,T) + n(t) \\ &= \sqrt{P_0T} \sum_{j=-\infty}^{\infty} a_j[g(t) \star h(t-jT)] + n(t) \\ &= \sqrt{P_0T} \sum_{j=-\infty}^{\infty} a_j p(t-jT) + n(t), \end{aligned} \tag{1.13}$$

where $\star$ denotes convolution and

$$v_{\boldsymbol{A}}(t,T) \stackrel{\triangle}{=} g(t) \star x_{\boldsymbol{A}}(t,T), \tag{1.14}$$

$$p(t) \stackrel{\triangle}{=} g(t) \star h(t). \tag{1.15}$$

The Fourier transform of $g(t) \star x_{\boldsymbol{A}}(t,T)$ is $G(f)X_{\boldsymbol{A}}(f,T)$, where $G(f)$ and $X_{\boldsymbol{A}}(f,T)$ are Fourier transforms of $g(t)$ and $x_{\boldsymbol{A}}(t,T)$, respectively. Thus, $g(t)$ changes the spectrum of $x(t)$, thereby being a *frequency selective* channel. If there is no multipath in the environment, then $g(t) = \delta(t)$, and no frequency selection occurs. In this case, $g(t)$ is a *non-frequency selective* channel, also known as a *flat* channel. In either case, it is further assumed that the receiver has perfect knowledge of $g(t)$.

1.1.3 Maximum-Likelihood Sequence Estimation

The optimal way to recover the data symbols $\boldsymbol{a}$ from $r(t)$ in (1.13) is by applying a maximum likelihood sequence estimation (MLSE) at the receiver. This amounts to solving the following optimization problem

$$\hat{\boldsymbol{a}} \stackrel{\triangle}{=} \arg\max_{\boldsymbol{a}} \Pr(r(t)|\boldsymbol{a}), \tag{1.16}$$

where $\Pr(r(t)|\boldsymbol{a})$ is the conditional probability density function (pdf) of $r(t)$ given that $\boldsymbol{a}$ is sent. It can be shown [9] that (1.16) is the optimal way to recover

$\boldsymbol{a}$ if and only if all possible symbol sequences $\boldsymbol{a}$ are equiprobable. Further, it is well-known [9] that the optimization (1.16), for AWGN channels, is equivalent to minimum Euclidean distance decoding

$$\begin{aligned}\hat{\boldsymbol{a}} &= \arg\min_{\boldsymbol{a}} \int_{-\infty}^{\infty} |r(t) - v_{\boldsymbol{a}}(t,T)|^2 \,\mathrm{d}t \\ &= \arg\min_{\boldsymbol{a}} \int_{-\infty}^{\infty} |r(t)|^2 - 2\,\mathrm{Re}\{r(t)v_{\boldsymbol{a}}^*(t)\} + |v_{\boldsymbol{a}}(t,T)|^2 \,\mathrm{d}t. \end{aligned} \quad (1.17)$$

Inserting the expression (1.14) for $v_{\boldsymbol{a}}(t,T)$, and noting that the term $\int |r(t)|^2 \,\mathrm{d}t$ has no impact on the minimization, (1.17) reduces to

$$\hat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} \sum_{j=0}^{\infty} \mathrm{Re}\{a_j^* y_j\} - \int_{-\infty}^{\infty} \frac{1}{2} |v_{\boldsymbol{a}}(t,T)|^2 \,\mathrm{d}t, \quad (1.18)$$

where

$$y_j \triangleq \int_{-\infty}^{\infty} r(t) p^*(t - jT) \,\mathrm{d}t. \quad (1.19)$$

The sequence $\boldsymbol{y} = \{\ldots, y_{-1}, y_0, y_1, \ldots\}$ can be obtained by applying a matched filter $p^*(-t)$ together with baud-rate sampling at the receiver. Furthermore, $\boldsymbol{y}$ is a *sufficient statistic* for detecting $\boldsymbol{a}$, i.e., knowing $\boldsymbol{y}$ is sufficient to perform MLSE. Expanding $r(t)$ as in (1.13), we get

$$\begin{aligned} y_j &= \sqrt{P_0 T} \sum_{k=-\infty}^{\infty} a_k \int_{-\infty}^{\infty} p(t - jT) p^*(t - kT) \,\mathrm{d}t + \int_{-\infty}^{\infty} n(t) p^*(t - jT) \,\mathrm{d}t \\ &= \sqrt{P_0 T} \sum_{k=-\infty}^{\infty} a_k z_{j-k} + \eta_j, \end{aligned} \quad (1.20)$$

with

$$z_{j-k} \triangleq \int_{-\infty}^{\infty} p(t - jT) p^*(t - kT) \,\mathrm{d}t \quad (1.21)$$

and

$$\eta_j \triangleq \int_{-\infty}^{\infty} n(t) p^*(t - jT) \,\mathrm{d}t. \quad (1.22)$$

The sequence $\boldsymbol{z} = \{\ldots, z_{-1}, z_0, z_1, \ldots\}$ is the *inter-symbol-interference* (ISI), and the noise sequence $\boldsymbol{\eta} = \{\ldots, \eta_{-1}, \eta_0, \eta_1, \ldots\}$ is a *colored* noise sequence if $z_k \neq 0$ for $k \neq 0$. Hence, a discrete time model of (1.13) is

$$\boldsymbol{y} = \sqrt{P_0 T}\boldsymbol{a} \star \boldsymbol{z} + \boldsymbol{\eta}. \quad (1.23)$$

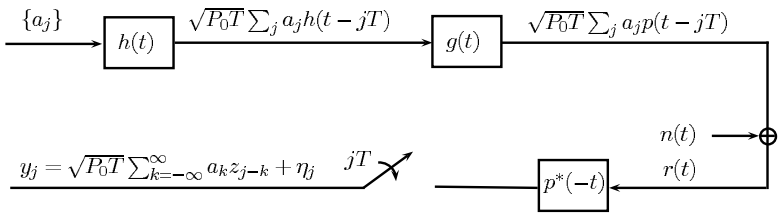

Figure 1.1: The analog communication model that gives rise to the discrete time model in (1.23).

Figure 1.1 shows the communication chain that gives rise to the discrete model in (1.23). An MLSE implementation on the model (1.13), as in (1.23), was proposed in [10], while an implementation over a whitened model was proposed in [11]. We have,

$$\begin{aligned} z_k &= \int_{-\infty}^{\infty} |P(f)|^2 e^{i2\pi kTf} \mathrm{d}f \\ &= \sum_{j=-\infty}^{\infty} \int_{-1/2T}^{1/2T} |P(f+j/T)|^2 e^{i2\pi kTf} \mathrm{d}f \\ &= \int_{-1/2T}^{1/2T} \sum_{j=-\infty}^{\infty} |P(f+j/T)|^2 e^{i2\pi kTf} \mathrm{d}f \\ &= \int_{-1/2T}^{1/2T} |P_{\mathrm{fo}}(f,T)|^2 e^{i2\pi kTf} \mathrm{d}f, \end{aligned} \tag{1.24}$$

where $|P_{\mathrm{fo}}(f,T)|^2$ is the *folded* pulse spectrum

$$|P_{\mathrm{fo}}(f)|^2 \stackrel{\triangle}{=} \sum_{j=-\infty}^{\infty} |P(f+j/T)|^2, \quad -1/2T \le f \le 1/2T. \tag{1.25}$$

Hence, applying a matched filter and sampling with frequency $1/T$ folds the spectrum of the received ISI sequence around $1/2T$: This is the well-known spectrum folding that occurs from sampling [9]. Note that $|P(f)|^2 = |H(f)|^2|G(f)|^2$, since $p(t)$ is a convolution of $h(t)$ and $g(t)$. For notational

convenience, we define

$$\begin{aligned} S_H(f) &\triangleq |H(f)|^2 \\ S_{HG}(f) &\triangleq |P(f)|^2 = |H(f)G(f)|^2 = S_H(f)S_G(f) \\ S_{H,\text{fo}}(f,T) &\triangleq |H_{\text{fo}}(f,T)|^2 \\ S_{HG,\text{fo}}(f,T) &\triangleq |P_{\text{fo}}(f,T)|^2. \end{aligned} \tag{1.26}$$

A graphical view of some of the spectra in (1.26) is illustrated in Figure 1.2.

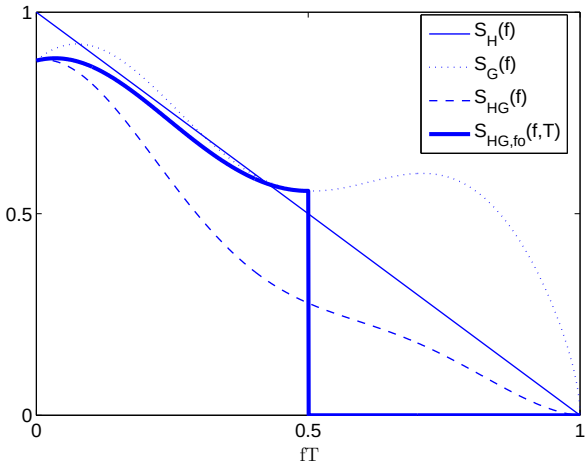

Figure 1.2: The quantities $S_H(f)$, $S_G(f)$, $S_{HG}(f)$ and $S_{HG,\text{fo}}(f,T)$. Note how folding occurs around the point $f = 1/2T$.

1.2 Information Rate of Single-Carrier Linear Modulation Systems

The information rate is a measure of how many bits that can be carried through a channel, and is of uttermost importance for communications. The channel of interest in Part I of this thesis is the AWGN channel in (1.13). This section derives the achievable information rates of this channel. First, we start by defining different information rates in Section 1.2.1 for a general AWGN channel

$y(t) = v(t) + n(t)$, where $v(t)$ is not constrained to a specific signaling form. Section 1.2.2 then finds closed form expressions of these rates for the model in (1.13).

1.2.1 Information rate and capacity

Let us first formally define what is meant by information rate for an analog transmission system in the baseband. Assume that k information bits, $\boldsymbol{b} = [b_0, \ldots, b_{k-1}]$, are to be transmitted across a communication channel. Each realization of these bits is encoded into certain analog waveforms $x_j(t)$, and transmitted across a channel with impulse response $g(t)$[3]. Assuming that the channel is bandlimited to W Hz, this also limits the PSD $S_X(f)$ of the analog analog waveform to a bandwidth of W Hz. At the receiver side, the waveform $y(t) = v(t) + n(t)$ is observed, where $n(t)$ is a random noise process and $v(t) = g(t) \star x(t)$. The receiver then performs low-pass-filtering, sampling and decoding in order to recover $\boldsymbol{b}$. This simple transmission system is depicted in Figure 1.3. Shannon showed [14] that a signal of bandwidth W Hz spans roughly $\approx 2WT$ independent dimensions during a time interval of T seconds. This means that a bandlimited signal of W Hz is completely specified by a set of $2WT$ numbers during T seconds, which can be viewed as coordinates in a $2WT$ dimensional space. Further, Shannon showed that these numbers can be put on time shifted sinc pulses, since sinc pulses are basis functions that span the space of analog signals. Thus, the whole analog signal can be viewed in a discrete way, where the discrete numbers specify the amplitudes of the sinc pulses that build up any signal $x(t)$. Hence, encoding the bit sequence $\boldsymbol{b}$ into a waveform $x(t)$ corresponds to encoding it into a symbol vector $\boldsymbol{x}$, which defines the amplitudes of the sinc pulses. During T seconds, roughly $2WT$ symbols $\boldsymbol{x} = [x_0, \ldots, x_{2WT-1}]$ are sent, and at the receiver, the $2WT$ coordinates $\boldsymbol{y} = [y_0, \ldots, y_{2WT-1}]$ that specify the received signal $y(t)$ are recovered (by means of low pass filtering and sampling). Let $\boldsymbol{X} = [X_0, \ldots, X_{2WT-1}]$ be a sequence of $2WT$ random variables and $p_{\boldsymbol{X}}(\boldsymbol{x})$ denote the pdf of the sequence. Note that $\boldsymbol{x}$ corresponds to a certain realization of $\boldsymbol{X}$, which specifies the signal $x(t)$. Similarly, let $\boldsymbol{Y} = [Y_0, \ldots, Y_{2WT-1}]$ denote the sequence of random variables observed at the receiver. A certain modulator specifies $p_{\boldsymbol{X}}$, and the fundamental question that arises is: How many bits per second can be transmitted across $g(t)$ with a certain modulator, so that they can be recovered without any error at the receiver? This rate is the *information rate*,

[3]Note that in general, the number of different possible waveforms $x_j(t)$ to choose from is larger than the number of possible bit sequences 2^k, i.e., the waveforms come from a larger set which also can be uncountable. To cleverly choose a subset of the waveforms to be used for transmission is the idea of coding.

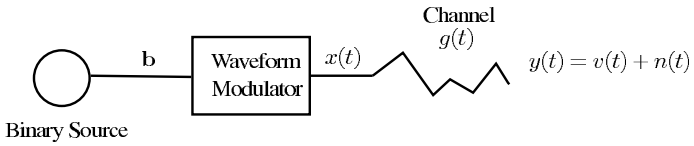

Figure 1.3: The communication chain for single carrier modulation. A binary source produces a bit sequence $\boldsymbol{b}$ of k bits which is encoded into an analog waveform $x(t)$. This waveform is then transmitted across a channel with impulse response $g(t)$, and a waveform $y(t) = v(t) + n(t)$ is observed at the receiver.

in bits per second, and can be achieved by cleverly choosing a subset of the possible waveforms $\{x_j(t)\}$. Shannon answered this question completely in [13, 14]. Morever, he even found the ultimate limit, i.e., the maximun that can be achieved. Thus, there exist ultimate limits on a communication system that can be computed. Let $\mathcal{I}(\boldsymbol{Y};\boldsymbol{X}) = \mathcal{H}(\boldsymbol{Y}) - \mathcal{H}(\boldsymbol{Y}|\boldsymbol{X})$ be the mutual information between the sequence $\boldsymbol{Y}$ and $\boldsymbol{X}$, where $\mathcal{H}(\cdot)$ is the differential entropy operator.

Definition 1. *The* information rate

$$I(p_{\boldsymbol{X}}) \stackrel{\triangle}{=} \lim_{T\to\infty} \mathcal{I}(\boldsymbol{Y};\boldsymbol{X})/T,$$

in bits per second, is the maximum number of information bits per second that can be carried with a fixed pdf $p_{\boldsymbol{X}}(\boldsymbol{x})$ across the channel $g(t)$.

Thus, given a certain pdf $p_{\boldsymbol{X}}$, it is in principle possible to calculate the upper limit on what can be achieved. Even though the information rate depends on the shape of $g(t)$, we will not explicitly write $g(t)$ as an argument of the information rate since this is understood implicitly. The information rate is the limit of systems with a fixed set of waveforms, but where one is free to choose a subset of them them to transmit. An example is that set of waveforms $\{x_j(t)\}$ is linear modulation with binary symbols, in which the subset to use is typically determined by the design of a code. If one maximizes the information rate with respect to the pdf $p_{\boldsymbol{X}}$, but keeps the PSD $S_X(f)$ constant, one obtains

Definition 2. *The* constrained capacity

$$C(S_X(f)) \stackrel{\triangle}{=} \sup_{p_{\boldsymbol{X}}:\mathrm{PSD}(\{x_j(t)\})=S_X(f)} I(p_{\boldsymbol{X}}))$$

is the maximum number of information bits per second that can be carried by a signal with PSD $S_X(f)$ across the channel $g(t)$.

Shannon showed in [14] that the constrained capacity is attained by a Gaussian distribution on the symbols $\{x_j\}$ in $\boldsymbol{x}$. Finally, maximizing the constrained capacity with respect to the average power constraint on the PSD $\int S_X(f)\,\mathrm{d}f \leq P_0$, one obtains

Definition 3. *The* capacity

$$\hat{C}(P_0) \triangleq \sup_{S_X(f):\int S_X(f)\,\mathrm{d}f \leq P_0} C(S_X(f))$$

is the maximum number of information bits per second that can be carried by an average power of P_0 across the channel $g(t)$.

Since the information rate, constrained capacity and capacity completely determine how much information that can be transmitted across a certain communication channel, it is thus of interest to compute these quantities.

1.2.2 Information rates for linear modulation systems

Let us now derive expressions for the rates in Section 1.2.1 for a general AWGN baseband signaling model $y(t) = g(t) \star x(t) + n(t)$, Shannon's classical results provide expressions for the constrained capacity and the capacity of it [14]. The constrained capacity for a certain PSD $S_X(f)$ of $x(t)$, in bits per second, equals

$$C(S_X(f)) \triangleq \int_0^W \log\left(1 + \frac{S_X(f)S_G(f)}{N_0}\right) \mathrm{d}f. \tag{1.27}$$

The rate in (1.27) can be achieved by a transmitted $x_{\boldsymbol{B}}(t,T)$ of the form in (1.2) and with $T = 1/2W$ [17],

$$x_{\boldsymbol{B}}(t, 1/2W) = \sqrt{P_0 T} \sum_{k=-\infty}^{\infty} b_k \mathrm{sinc}(t - k/2W), \tag{1.28}$$

where $\{b_k\}$ is a sequence of complex-valued Gaussian data symbols and $\mathrm{sinc}(t)$ is the sinc pulse. Since the pulse is a sinc, the PSD of $x_{\boldsymbol{B}}(t, 1/2W)$ is $P_0 S_{\boldsymbol{B}}(f, 1/2W)$ in the bandlimited interval $[-W, W]$, thus, the correlation of $\{b_k\}$ determines the PSD, and it is chosen such that the PSD constraint $P_0 S_{\boldsymbol{B}}(f, 1/2W) = S_X(f)$ is satisfied. Hence, linear modulation with Gaussian data symbols can achieve the constrained capacity in (1.27), by signaling with Gaussian data symbols and sinc pulses at the signaling rate $1/T = 2W$. In the case of a flat channel, detection of the symbols $\{b_k\}$ in (1.28) is simple, since they are put on the zero-crossings of the sinc pulses. Another possibility to

achieve (1.28) is to have uncorrelated Gaussian symbols $\{b_k\}$ and a non-sinc pulse $h(t)$, as in (1.2), such that the PSD is still $S_X(f)$.

Optimizing (1.27) over $S_X(f)$, subject to the average power constraint $\int S_X(f)\,\mathrm{d}f \leq P_0$, gives the capacity of the AWGN channel

$$\begin{aligned}\hat{C}(P_0) &\stackrel{\triangle}{=} \max_{S_X(f):\int S_X(f)\,\mathrm{d}f \leq P_0} C(S_X(f)) \\ &= \int_0^W \log\left(\max\left(\theta(P_0)\frac{S_G(f)}{N_0}, 1\right)\right)\mathrm{d}f \\ &\qquad \text{subject to} \\ &\int_0^W \max\left(\theta(P_0) - \frac{N_0}{S_G(f)}, 0\right)\mathrm{d}f = P_0.\end{aligned} \tag{1.29}$$

$\hat{C}(P_0)$ is the maximal achievable information rate over the channel with impulse response $g(t)$, under an average transmission power P_0. The optimal $S_X(f)$ which gives $\hat{C}(P_0)$ obeys the well-known waterfilling policy, where $\theta(P_0)$ is a real scalar value that represents the waterfilling level (i.e., fulfills the second integral equation). As before, (1.29) is achieved by signals of the form in (1.28), where $\boldsymbol{B} = \{b_k\}$ are such that the optimal PSD is obtained.

As discussed above, rates in (1.27) and (1.29) are achievable by linear modulation signals of the form in (1.2), with a signaling rate of $1/T = 2W$. Let us now discuss what rates that are achievable by (1.2) for a fixed T which may be less than $1/2W$. Note that the signaling rate $1/T$ in (1.2) has no impact on the transmitted PSD $S_{X_{\boldsymbol{A}}}(f,T)$ of $x_{\boldsymbol{A}}(t)$, i.e., $S_{X_{\boldsymbol{A}}}(f,T) = S_{X_{\boldsymbol{A}}}(f,T')$ for any T', since different time shifts of $h(t)$ only shift the frequency components through a complex exponential, and this has no impact on the PSD. Thus, with the constraint $S_{X_{\boldsymbol{A}}}(f,T) = S_X(f)$ on the PSD, the rate in (1.27) is the upper limit, achievable by signaling with $T = 1/2W$ and Gaussian symbols $\{a_j\}$ that give rise to the PSD $S_X(f)$. However, if the signaling rate $1/T$ in (1.2) does not equal $2W$, it is not clear whether (1.27) and (1.29) can be achieved anymore, even though the transmitted PSD constraint is still met. In order to gain insight into what happens with different signaling rates, it is sufficient to study the discrete time model in (1.23), which is lossless from an information rate point of view due to the fact that the sequence $\{y_j\}$ is a sufficient statistic. Hence, the achievable rates for (1.2) with different $1/T$ can be found by calculating the information rate of (1.23), which will soon be done. Before doing that, we note that the discrete time model in (1.2) changes considerably if the signaling rate is varied. Namely, it follows from (1.24) that the ISI sequence

$\{z_k\}$ directly depends on T, and the spectrum of the ISI changes with T since it depends on the folded spectrum in (1.25). Thus varying the signaling rate in (1.2) changes the ISI sequence and its spectrum, and it is not obvious how this impacts the information rate of (1.23). If the spectrum of the ISI sequence affects the constrained capacity of (1.23), then this might give rise to a significant change in the achievable information rates of (1.23). Hence, we need to calculate the achievable rates for (1.23) and investigate their dependence on the signaling rate $1/T$.

In practical single carrier systems, the symbols $\{a_j\}$ in (1.2) are taken from a discrete alphabet such as quadrature amplitude modulation (QAM). In this case, the analog waveforms are not built up from Gaussian symbols, and thus the maximum number of carried bits per second is given by the information rate in Definition 1. Hence, the bit rates given by the constrained capacity and the capacity are not achievable with discrete alphabets[4]. Moreover, for a discrete alphabet, it is not possible to obtain a simple closed form expression for the information rate. However, it turns out that for large QAM alphabets, the behaviour of the information rate is very well predicted by the information rate for Gaussian symbols, i.e., the constrained capacity. Since there exists an expression for the constrained capacity and the capacity of the AWGN channel, we thus henceforth assume Gaussian distributed symbols.

Let

$$Z(\lambda, T) = \sum_k z_k e^{i\lambda k} \tag{1.30}$$

be the Fourier transform of the sequence $\boldsymbol{z}$, here given in angular frequency. $Z(\lambda, T)$ depends on T through the sequence $\{z_j\}$, which in turn depends on T through (1.24). Similarly, $S_{\boldsymbol{A}}(\lambda, T)$ denotes the the angular frequency expression of $S_{\boldsymbol{A}}(f, T)$. The constrained capacity of (1.23) is given by [16, 17, 18]

$$C(P_0 S_{\boldsymbol{A}}(f,T), T) = \frac{1}{2\pi T} \int_0^{\pi} \log_2 \left(1 + \frac{P_0 S_{\boldsymbol{A}}(\lambda, T) Z(\lambda, T)}{N_0}\right) \mathrm{d}\lambda. \tag{1.31}$$

Since the constrained capacity now depends on the signaling rate T through $Z(\lambda, T)$, we add T as one of its arguments. From the expression for $\{z_j\}$ in (1.21), it can be shown that

$$\begin{aligned} Z(\lambda, T) &= \frac{1}{T} \sum_{k=-\infty}^{\infty} \left| P\left(\frac{\lambda}{2\pi T} + \frac{k}{T}\right)\right|^2 = \frac{1}{T} \left| P_{\text{fo}}\left(\frac{\lambda}{2\pi T}\right)\right|^2 \\ &= \frac{1}{T} S_{HG,\text{fo}}\left(\frac{\lambda}{2\pi T}, T\right). \end{aligned} \tag{1.32}$$

[4]We will later see, however, that $I(p_{\boldsymbol{X}}) \to C(S_X(f))$ as $T \to 0$ even with binary inputs.

Hence, $Z(\lambda, T)$ is proportional to the folded spectrum of $S_{HG}(f)$ around the frequency $f = \lambda/2\pi T$. Inserting (1.32) into (1.31), and performing a change of variables, we get

$$C(P_0 S_{\boldsymbol{A}}(f,T),T) = \int_0^{1/2T} \log_2\left(1 + \frac{P_0 S_{\boldsymbol{A}}(f,T) S_{HG,\mathrm{fo}}(f,T)}{N_0}\right) \mathrm{d}f. \quad (1.33)$$

Hence, the constrained capacity of (1.23) clearly depends on the folded spectrum $S_{HG,\mathrm{fo}}(f,T)$, and since the latter changes with the signaling rate, (1.33) can also change with the signaling rate. Note that the constrained capacity also depends on the pulse spectrum $S_H(f)$, but again, this dependence will not be explicitly written out, since in most cases the pulse shape is fixed. Maximizing the constrained capacity (1.33) over $S_{\boldsymbol{A}}(f,T)$, subject to the constraint on $S_{\boldsymbol{A}}(f,T)$ in (1.10), gives rise to the capacity of (1.23), where we again remind the reader that this capacity is constrained on T. The maximizing $S_{\boldsymbol{A}}(f,T)$ obeys the waterfilling (WF) strategy and gives

$$\begin{aligned} \hat{C}(P_0,T) &\triangleq \max_{S_{\boldsymbol{A}}(f,T)} C(P_0 S_{\boldsymbol{A}}(f,T),T) \\ &= \int_0^{1/2T} \log\left(\max\left(\theta(P_0,T)\frac{S_{HG,\mathrm{fo}}(f,T)}{N_0 S_{H,\mathrm{fo}}(f,T)}, 1\right)\right)\mathrm{d}f \\ &\qquad \text{subject to} \\ &\int_0^{1/2T} \max\left(\theta(P_0,T) - \frac{N_0 S_{H,\mathrm{fo}}(f,T)}{S_{HG,\mathrm{fo}}(f,T)}, 0\right)\mathrm{d}f = P_0. \end{aligned} \quad (1.34)$$

$\theta(P_0,T)$ is a real scalar value that represents the waterfilling level (i.e., fulfills the second integral equation). Comparing the equations in (1.33) and (1.34) with (1.27) and (1.29), the difference is that folding of the spectra occurs in the former.

1.3 Impact of Signaling Rate on the Information Rate for Single-Carrier Linear Modulation Systems

To recap the results in Section 1.2.2, transmitting symbols by means of (1.2) every Tth second, and applying a matched filter as in (1.19), gives rise to a discrete model in (1.23) with the constrained capacity $C(P_0 S_{\boldsymbol{A}}(f,T),T)$ in (1.27)

and capacity $\hat{C}(P_0, T)$ in (1.29). The rates in (1.27) and (1.29) are independent on the receiver structure, thus they are the upper limits on the achievable rate of (1.23). However, in (1.23), folded spectra $S_{H,\text{fo}}(f, T)$ and $S_{HG,\text{fo}}(f, T)$ occur, which have direct impact on the achievable rates of (1.23). The reason for this is the sampling rate, $1/T$, which gives a different ISI sequence $\boldsymbol{z}$ in (1.21). Its transform in (1.32) is now a folding of $S_{HG}(f)$ around $f = \lambda/2\pi T$, which clearly depends on the signaling rate $1/T$. Therefore, different signaling rates can vary the constrained capacity (we will soon see that this is indeed the case). In general, the higher the baud rate, the more severe ISI sequence $\boldsymbol{z}$ results at the receiver (i.e., longer ISI sequence), which gives rise to a large MLSE decoding complexity.

Although massive research has been conducted on single carrier linear modulation during the past 90 years since Nyquist's seminal paper [12], the exact relation between the capacity changes and signaling rate changes remains unknown. Recently, [15] showed that binary amplitude modulation with infinite signaling rate achieves the constrained capacity, hence, binary transmission is lossless if the signaling rate approaches infinity. However, for finite signaling rates the behavior of the capacities is still unknown. Part I studies the exact change in the constrained capacity and the capacity for small changes of the signaling rate. As will be seen, the outcome is not unique. Under certain conditions, these rates can never be degraded by increasing the signaling rate. However, under other conditions, they can in fact decrease when the signaling rate is increased.

In this thesis, we model the change in signaling rate by inserting a real-valued number $0 < \tau \leq 1$ in front of T, so that the transmitted signal in (1.2) now becomes

$$x_{\boldsymbol{A}}(t, \tau T) = \sqrt{P_0 \tau T} \sum_{j=-\infty}^{\infty} a_j h(t - j\tau T). \tag{1.35}$$

Hence, $1/T$ is a reference signaling rate, and we are free to increase it as much as desired by lowering τ. All previously introduced functions have the same expressions, except that T is now replaced with τT which makes them dependent on τ. Thus, the constrained capacity $C(P_0 S_{\boldsymbol{A}}(f, T), T)$ is now denoted as $C(P_0 S_{\boldsymbol{A}}(f, \tau T), \tau T)$, and similarly for all other functions depending on T.

In [18], the impact of τ on the constrained capacity was analyzed to a large extent for flat channels ($G(f) = 1$) and unit energy T-orthogonal pulses $h(t)$, i.e.,

$$\int h(t) h^*(t - kT)\, \mathrm{d}t = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0. \end{cases} \tag{1.36}$$

The T-orthogonality for flat channels results in an ISI sequence in (1.23) such that $z_k = \delta(k)$ for $\tau = 1$, and thus detection under T-orthogonal signaling

rate is very simple. Also the information rate $I(p_{\boldsymbol{X}})$ with QAM inputs was studied in [18] by means of simulation, since no closed form exists for $I(p_{\boldsymbol{X}})$ with discrete inputs. Note that from (1.32), $z_k = \delta(k)$ implies that folding of $S_{HG,\text{fo}}\left(\frac{\lambda}{2\pi T}\right) = S_{H,\text{fo}}(f)$ at the frequency $1/2T$ gives rise to a flat spectrum, which is the frequency domain equivalent of the constraint in (1.36). This signaling rate is called the *orthogonal signaling* rate. A further assumption in [18] was that $S_{\boldsymbol{A}}(f,T) = 1$, i.e., the data symbols are uncorrelated. Hence, the constrained capacity is now denoted as $C(P_0, \tau T)$. Under these assumptions, the following theorem was proved in [18].

Theorem 1. *Unless $h(t)$ is a sinc pulse, for $\tau = 1/N$, N an integer, it holds that $C(P_0, \tau T) > C(P_0, T)$.*

Hence, by increasing the signaling rate above $1/T$ for T-orthogonal pulses, it is possible to achieve a higher constrained capacity than with orthogonal signaling. This concept is known in the literature as *faster-than-Nyquist signaling*, since Nyquist signaling corresponds to an orthogonal signaling rate[5]. The main idea behind the proof of Theorem 1 is that for a τ such that $1/\tau T = 2W$, i.e., by signaling at the rate $1/2W$, no folding of the spectrum occurs in (1.33). Thus, the original spectrum $S_{V_{\boldsymbol{A}}}(f, \tau T)$, which equals $S_H(f)$ in this case since $S_{\boldsymbol{A}}(f, \tau T) = 1$ and $S_G(f) = 1$, is recovered and the information rate in (1.33) equals the maximal one in (1.27). Instead, signaling at the lower orthogonal rate of $1/T$, the expression in (1.33) is strictly lower than (1.27), except in the cases when $h(t)$ is a sinc pulse bandlimited to $1/2T$ positive Hz.

In the rest of Part I we shall study the quantities

$$\frac{\partial C(P_0 S_{\boldsymbol{A}}(f, \tau T), \tau T)}{\partial \tau}$$

and

$$\frac{\partial \hat{C}(P_0, \tau T)}{\partial \tau}.$$

In particular, we seek to investigate when the derivative is negative, i.e., when it is beneficial to signal faster. For convenience we restrict W to satisfy $1/2T \leq W \leq 1/T$. This choice is made since it allows us to make use of the simpler Nyquist criterion for T-orthogonal pulses, $S_H(f) + S_H(1/T - f) = T$, $0 \leq f \leq 1/2T$, instead of the more clumsy Gibby-Smith condition $S_{H,\text{fo}}(f) = T$ [19]. There are several different cases to consider which can be summarized into

- Case I: $S_{\boldsymbol{A}}(f, \tau T) = 1$, $S_H(f)$ fixed for all τ.

[5] This is the main reason for the T-orthogonal assumption on $h(t)$ in (1.4): Being able to make information rate comparisons to fully orthogonal transmissions for flat channels. However, the results in this thesis are more general and do not require this assumption.

- Case II: $S_{\boldsymbol{A}}(f, \tau T)$ free, $S_H(f)$ fixed for all τ.
- Case III: $S_H(f)$ free to choose for every τ.

With "$S_H(f)$ free", we mean a scenario where the transmitter is free to choose the spectrum shaping filter at will (possibly from a large filter bank). With "$S_{\boldsymbol{A}}(f, \tau T)$ free" we mean that the transmitter can apply the waterfilling technique to optimize the information rate, i.e., in that case we study the derivative of (1.34). Case I corresponds to the most important case from a practical point of view, namely that uncorrelated data is assumed and the transmitter has a fixed spectrum shaping filter (pulse).

Due to the bandlimitation of $S_H(f)$, it follows that if we choose $\tau < 1/(2WT)$, no folding of the spectrum $S_{HG}(f)$ occurs. Hence, for $\tau < 1/(2WT)$ we have

$$S_{HG,\mathrm{fo}}(f, \tau T) = S_H(f) S_G(f).$$

Inserting this into (1.33) gives

$$C(P_0 S_{\boldsymbol{A}}(f, \tau T), \tau T) = \int_0^W \log_2 \left(1 + \frac{P_0\, S_{\boldsymbol{A}}(f, \tau T)\, S_H(f)\, S_G(f)}{N_0}\right) \mathrm{d}f, \quad (1.37)$$

which equals the maximum in (1.27) with $S_X(f) = P_0 S_{\boldsymbol{A}}(f, \tau T) S_H(f) S_G(f)$. With $S_{\boldsymbol{A}}(f, \tau T) = 1$, (1.37) is independent of τ, as long as $\tau < 1/(2WT)$. For "$S_{\boldsymbol{A}}(f, \tau T)$ free"', $\tau < 1/(2WT)$ implies that the optimal $S_{\boldsymbol{A}}(f, \tau T)$ concentrates its power to the frequency range where $S_H(f)$ is non-zero, and thus (1.29) is achievable. Hence, there is no need to ever consider values of τ that are smaller than $1/(2WT)$, so the interesting regime of τ is $\tau \geq 1/(2WT)$.

1.4 Analysis of the Constrained Capacity

Throughout, the following notation will be used for derivatives: $f'(t)$ is the derivative of a one-variable function $f(t)$ and $f''(t)$ is the second derivative of $f(t)$. For a function $f(x_1, x_2)$ of two variables, $f'_{x_1}(x_1, x_2)$ denotes the derivative with respect to x_1, if it is not clear with regard to which variable the derivation is performed.

1.4.1 $S_{\boldsymbol{A}}(f, \tau T) = 1$, $H(f)$ fixed for all τ

We start our analysis for this case by assuming a frequency non-selective channel ($G(f) = 1$). At the end of this section, the assumption of a flat channel

will be relaxed and the derived results also apply to frequency selective channels. Since $h(t)$ is T-orthogonal, the ISI will now be incurred by choosing a non-orthogonal signaling rate (i.e. $\tau < 1$) in (1.2). When $\tau = 1$ (no ISI), it follows that $S_{H,\mathrm{fo}}(f, \tau T) = T$.

Since $S_{\boldsymbol{A}}(f, \tau T) = 1$ and $S_G(f) = 1$, (1.33) is

$$C\left(P_0, \tau T\right) = \int_0^{\frac{1}{2T\tau}} \log_2 \left(1 + \frac{P_0}{N_0} S_{H,\mathrm{fo}}(f, \tau T)\right) \mathrm{d}f. \tag{1.38}$$

For Nyquist signaling with a T-orthogonal $h(t)$ ($\tau = 1$), the constrained capacity $C\left(P_0, \tau T\right)$ equals the familiar

$$C_N(P_0) \triangleq C\left(P_0, T\right) = \frac{1}{2T} \log_2(1 + 2P_0T/N_0). \tag{1.39}$$

We will refer to $C_N(P_0)$ as the Nyquist information rate. It does not depend on the pulse $h(t)$, as long as $h(t)$ is T-orthogonal and unit-energy. Similarly, $C_N(P_0)$ depends only on the T-orthogonality rate, and not on signaling rate variations with varying τ.

The τ interval of interest for the flat channel case is $1/(2WT) \leq \tau \leq 1$. That $\tau > 1$ is a loss can be realized as follows. First, it can be observed that the integration interval in (1.38) is smaller when $\tau > 1$ compared to $\tau \leq 1$. Secondly, (1.38) is maximized for a flat shape on $S_{H,\mathrm{fo}}(f, \tau T)$; for $\tau = 1$, this maximum is achieved since $S_{H,\mathrm{fo}}(f, \tau T)$ becomes flat due to Nyquist's "1924" criteria for T-orthogonal pulses. Combining these two observations leads to the fact that $C\left(P_0, \tau T\right) < C(P_0, T)$ for $\tau > 1$.

In principle three different behaviors can be imagined for the constrained capacity of a flat channel. It can be monotonically increasing with decreasing τ, i.e. $C_\tau^{'}(P_0, \tau T) < 0$; this is behavior 1 in Figure 1.4 (which is generated with $W = 1/T$). Another behavior that could be imagined is that $C\left(P_0, \tau T\right)$ is not monotonically increasing with decreasing τ; this is behavior 2. The third behavior is a curve that goes below the starting value $C_N(P_0)$. This is behavior 3 in Figure 1.4. Note that the curves reach a maximum at $\tau = 0.5$ since no folding occurs for $\tau \leq 0.5$. Sufficient conditions for the pulse $h(t)$ will be derived for which $C\left(P_0, \tau T\right)$ satisfies one of the three different behaviors. Note that behavior 1 is well defined also for frequency selective channels: Is the constrained capacity monotonically increasing or not?

First, behavior 1 is analyzed, which asks for pulses for which $C\left(P_0, \tau T\right)$ increases monotonically with decreasing τ, that is, whether there exist pulses that have $C_\tau^{'}(P_0, \tau T) < 0$ for $1/(2WT) < \tau < 1$.

In what follows it is assumed that $S_H(f)$ is continuous in $f \in [0, W]$, while $S_H^{'}(f)$ is discontinuous in at most a finite number of points in the interval.

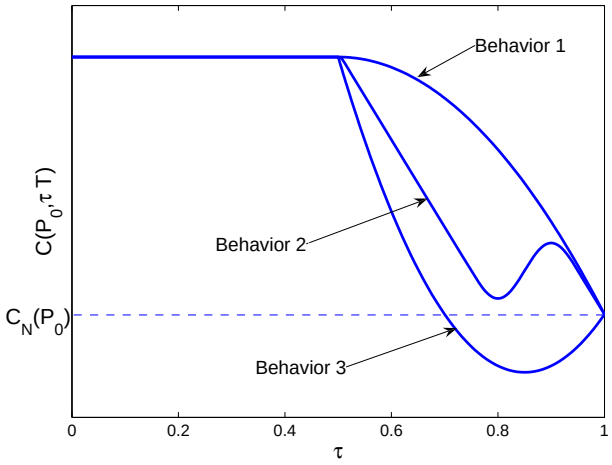

Figure 1.4: Three different behaviors for $C\left(P_0, \tau T\right)$ with respect to τ.

First, the results of the analysis are summarized, while practical examples are given in Section 1.4.2.

Theorem 2. *Assume that $S_H(f)$ is decreasing in $[0, W]$, that the smallest value of $S_{H,\mathrm{fo}}(f, \tau T)$ in the interval $f \in \left[\frac{1}{\tau T} - W, 1/2\tau T\right]$ occurs at $f = 1/2\tau T$ and that $S_{H,\mathrm{fo}}(f, \tau T)$ is not identically zero in that interval. Then $C\left(P_0, \tau T\right)$ is monotonically increasing for decreasing τ and larger than $C_N(P_0)$ for $\tau < 1$. If $S_{H,\mathrm{fo}}(f, \tau T) = 0$ for $f \in \left[\frac{1}{\tau T} - W, 1/2\tau T\right]$, then $C\left(P_0, \tau T\right)$ is non-decreasing for decreasing τ.*

Proof. We investigate for which pulses it is true that $C(P_0, \tau T)$ increases with decreasing τ. We study the sign of the derivative of $C(P_0, \tau T)$ in order to answer this question. For increased readability, we will in all proofs use ln instead of $\log_2$. This has no impact on the derived results since we are only interested in the sign of the derivative. Consider a pulse with a continuously differentiable spectrum in the frequency interval $[0, W]$. If the spectrum is discontinuous at $f = W$, we assume that the pulse values approaching the point from left and right are finite and positive. In the folded pulse shape, at the point $f = \frac{1}{\tau T} - W$, we might have a discontinuity in the value of the integrand in $C(P_0, \tau T)$ and its derivative with respect to τ. Therefore, we split

$C(P_0, \tau T)$ into two integrals:

$$\begin{aligned} C(P_0,\tau T) &= \lim_{\epsilon\to 0} \int_0^{\frac{1}{\tau T}-W-\epsilon} \ln\left(1+\frac{2P_0}{N_0}S_{H,\mathrm{fo}}(f,\tau T)\right)\mathrm{d}f \\ &\quad + \lim_{\epsilon\to 0} \int_{\frac{1}{\tau T}-W+\epsilon}^{1/2\tau T} \ln\left(1+\frac{2P_0}{N_0}S_{H,\mathrm{fo}}(f,\tau T)\right)\mathrm{d}f \end{aligned} \tag{1.40}$$

In the first integral, we approach the discontinuity point $f = \frac{1}{\tau T} - W$ from left and note that the integrand is bounded and well-defined for that limit, since the pulse spectrum also is. The same holds when approaching the discontinuity from the right, in the second integral. Both integrands are continuously differentiable in respective (open) integration intervals with respect to f and τ, and the first does not depend on τ. From this we get that $C(P_0, \tau T)$ is a differentiable function with respect to τ. This allows application of the Leibniz integral rule on $C(P_0, \tau T)$ [20]. The Leibniz integral rule states that for functions $f(x, z)$, $a(z)$ and $b(z)$,

$$\frac{\partial}{\partial z}\int_{a(z)}^{b(z)} f(x,z)\,\mathrm{d}x = \int_{a(z)}^{b(z)} \frac{\partial f(x,z)}{\partial z}\,\mathrm{d}x + f(b(z)),z)\frac{\partial b(z)}{\partial z} - f(a(z),z)\frac{\partial a(z)}{\partial z}. \tag{1.41}$$

Applying (1.41) on $C(P_0, \tau T)$ with respect to τ gives the following expression:

$$\begin{aligned} C'_\tau(P_0,\tau T) &= \ln\left(1+\frac{2P_0}{N_0}S^+_{H,\mathrm{fo}}\left(\frac{1}{\tau T}-W,\tau T\right)\right)\left(-\frac{1}{\tau^2 T}\right) \\ &\quad + \ln\left(1+\frac{2P_0}{N_0}S_{H,\mathrm{fo}}(1/2\tau T,\tau T)\right)\left(-\frac{1}{2\tau^2 T}\right) \\ &\quad + \ln\left(1+\frac{2P_0}{N_0}S^-_{H,\mathrm{fo}}\left(\frac{1}{\tau T}-W,\tau T\right)\right)\left(\frac{1}{\tau^2 T}\right) \\ &\quad + \lim_{\epsilon\to 0} \int_{\frac{1}{\tau T}-W+\epsilon}^{1/2\tau T} \frac{\frac{2P_0}{N_0}S'_H\left(\frac{1}{\tau T}-f\right)}{1+\frac{2P_0}{N_0}S_{H,\mathrm{fo}}(f,\tau T)}\left(-\frac{1}{\tau^2 T}\right)\mathrm{d}f \end{aligned} \tag{1.42}$$

where S^+_H and S^-_H denote limits from the right and left. In the case when we have several discontinuity points in the interval $(0, W]$, we split the integral as above at the discontinuity points and do similar calculations. This leads to more complicated expressions, which are not included here. Consider now

pulse spectra that are continuous at $f = W$, i.e., $S_H(W) = 0$. In that case, the derivative (1.42) reduces to

$$\begin{aligned} C_\tau^{'}(P_0, \tau T) &= \ln\left(1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(1/2\tau T, \tau T)\right)\left(-\frac{1}{2\tau^2 T}\right) \\ &\quad + \lim_{\epsilon \to 0} \int_{\frac{1}{\tau T} - W + \epsilon}^{1/2\tau T} \frac{\frac{2P_0}{N_0} S_H^{'}\left(\frac{1}{\tau T} - f\right)}{1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(f, \tau T)}\left(-\frac{1}{\tau^2 T}\right) \mathrm{d}f. \end{aligned} \tag{1.43}$$

From (1.43), we can finish the proof. We prove that the derivative in (1.43) is smaller than or equal to 0. We have that

$$\begin{aligned} &\lim_{\epsilon \to 0} \int_{\frac{1}{\tau T} - W - \epsilon}^{1/2\tau T} \frac{\frac{2P_0}{N_0} S_H^{'}\left(\frac{1}{\tau T} - f\right)}{1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(f, \tau T)}\left(-\frac{1}{\tau^2 T}\right) \mathrm{d}f \\ &\leq \lim_{\epsilon \to 0} \int_{\frac{1}{\tau T} - W - \epsilon}^{1/2\tau T} \frac{\frac{2P_0}{N_0} S_H^{'}\left(\frac{1}{\tau T} - f\right)}{1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(1/2\tau T, \tau T)}\left(-\frac{1}{\tau^2 T}\right) \mathrm{d}f, \end{aligned}$$

because $S_H(f)$ is decreasing in $[1/2\tau T, W)$ and the smallest value of $S_{H,\text{fo}}(f, \tau T)$ in the interval $\left(\frac{1}{\tau T} - W, 1/2\tau T\right]$ is $S_{H,\text{fo}}(1/2\tau T, \tau T) = 2S_H(1/2\tau T)$. Now

$$\begin{aligned} &\lim_{\epsilon \to 0} \int_{\frac{1}{\tau T} - W - \epsilon}^{1/2\tau T} \frac{\frac{2P_0}{N_0} S_H^{'}\left(\frac{1}{\tau T} - f\right)}{1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(1/2\tau T, \tau T)}\left(-\frac{1}{\tau^2 T}\right) \mathrm{d}f \\ &= \frac{\frac{2P_0}{N_0} S_H(1/2\tau T)}{\tau^2 T(1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(1/2\tau T, \tau T))}. \end{aligned}$$

Also

$$\frac{\frac{2P_0}{N_0} S_H(1/2\tau T)}{\tau^2 T\left(1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(1/2\tau T, \tau T)\right)} \tag{1.44}$$

$$\leq \ln\left(1 + \frac{2P_0}{N_0} S_{H,\text{fo}}(1/2\tau T, \tau T)\right)\left(\frac{1}{2\tau^2 T}\right) \tag{1.45}$$

which reduces to

$$\frac{\frac{4P_0}{N_0} S_H(1/2\tau T)}{1 + \frac{4P_0}{N_0} S_H(1/2\tau T)} \leq \ln\left(1 + \frac{4P_0}{N_0} S_H(1/2\tau T)\right) \tag{1.46}$$

and this is true for $\frac{4P_0}{N_0} S_H(1/2\tau T) \geq 0$. Hence, it follows that

$$\begin{aligned} C_\tau^{'}(P_0, \tau T) &\leq \frac{\frac{2P_0}{N_0} S_H(1/2\tau T)}{\tau^2 T(1 + \frac{2P_0}{N_0} S_{H,\mathrm{fo}}(1/2\tau T, \tau T))} \\ &\quad + \ln\left(1 + \frac{2P_0}{N_0} S_{H,\mathrm{fo}}(1/2\tau T, \tau T)\right)\left(-\frac{1}{2\tau^2 T}\right) \\ &\leq 0. \end{aligned} \tag{1.47}$$

In the case when $S_{H,\mathrm{fo}}(f, \tau T)$ is not identically 0 in $\left[\frac{1}{\tau T} - W, 1/2\tau T\right]$, there is strict inequality in the first step in the proof above, which implies that the derivative is strictly smaller than 0. This proves the theorem. □

From Theorem 2, we deduce the following corollary.

Corollary 1. *Assume that $S_H(f)$ is a decreasing function in $[0, W]$. If $S_H^{''}(f) \leq 0$ for $f \in [0, 1/2T]$, then $C(P_0, \tau T)$ is non-decreasing for decreasing τ.*

Proof. It is enough to prove that the smallest value is at $f = 1/2\tau T$ for some fixed τ. Nyquist orthogonality criteria $S_H(f) + S_H(1/T - f) = T$ gives $S_H^{''}(f) + S_H^{''}(1/T - f) = 0$. This gives that $S_H^{''}(f) \geq 0$ for $f \in [1/2T, 1/T]$ since $S_H^{''}(f) \leq 0$ for $f \in [0, 1/2T]$. From this it follows that $S_{H,\mathrm{fo}}^{'}(f, \tau T) = S_H^{'}(f) - S_H^{'}(1/\tau T - f) \leq 0$ for $f \in [0, 1/2\tau T]$, with equality when $f = 1/2\tau T$, which proves the corollary. □

Hence, Theorem 2 gives a simple condition on a pulse spectrum $S_H(f)$ that is sufficient for the constrained capacity to increase with the signaling rate. It is simply a matter of locating the minimum value of the folded pulse spectrum. Furthermore, Corollary 1 shows that if $S_H(f)$ is a decreasing concave function, then the constrained capacity increases with the signaling rate.

We next consider behavior 2. Assume that $S_H(f)$ is at most discontinuous at a finite number of points in the interval $f \in [0, W]$. We can prove

Theorem 3. *If $S_{H,\mathrm{fo}}(f, \tau T) \leq T$ in $[0, 1/2\tau T]$ then $C(P_0, \tau T) > C_N(P_0)$. Moreover, $C(P_0, \tau T) - C_N(P_0)$ is monotonically increasing with P_0 for any choice of τ.*

Proof. To simplify notation, we put $\xi = 2P_0/N_0$. Define

$$g(\xi) = \int_0^{1/2\tau T} \ln(1 + \xi S_{H,\mathrm{fo}}(f, \tau T))\mathrm{d}f - \frac{1}{2T}\ln(1 + \xi T). \tag{1.48}$$

Without loss of generality, it can be assumed that the integrand is a continuously differentiable function with respect to ξ and f. If it is discontinuous in at most finitely many f, the integral is split into intervals where the integrand is continuous. Hence, differentiation under the integral sign is allowed. It holds that $g(0) = 0$ and

$$g'(\xi) = \int_0^{1/2\tau T} \frac{S_{H,\mathrm{fo}}(f,\tau T)}{1+\xi S_{H,\mathrm{fo}}(f,\tau T)}\mathrm{d}f - \frac{1}{2(1+T\xi)}. \tag{1.49}$$

Hence $g'(0) = 0$, since $\int_0^{1/2\tau T} S_{H,\mathrm{fo}}(f,\tau T)\mathrm{d}f = 1/2$ because $h(t)$ has unit energy. From (1.49) we infer that

$$g'(\xi) > \int_0^{1/2\tau T} \frac{S_{H,\mathrm{fo}}(f,\tau T)}{1+\xi T}\mathrm{d}f - \frac{1}{2(1+\xi T)} = 0, \tag{1.50}$$

since $S_{H,\mathrm{fo}}(f,\tau T) \leq T$ with strict inequality in some interval. Since $g(\xi)$ and $g'(\xi)$ are continuous, we infer from above that $g(\xi) > 0$ when $\xi > 0$, which proves the theorem. □

Hence, Theorem 3 presents a simple condition on $S_{H,\mathrm{fo}}(f,\tau T)$ that gives a superior constrained capacity than the Nyquist information rate. Moreover, for such $S_{H,\mathrm{fo}}(f,\tau T)$, Theorem 3 shows that increasing the power P_0 widens the gap to the Nyquist information rate. However, it might happen that $C'_\tau(P_0,\tau T) > 0$ for some τ, i.e., the constrained capacity can decrease (but never below $C_N(P_0)$). From Theorem 3 we deduce the following corollary.

Corollary 2. *Assume that $h(t)$ is T-orthogonal. If $S_H(f)$ is decreasing in $[0,W]$ then $C(P_0,\tau T) - C_N(P_0)$ is monotonically increasing with P_0.*

Proof. Since $h(t)$ is T-orthogonal and $S_H(f)$ is decreasing in $[0,W]$ we have $T = S_H(f) + S_H(1/T - f) \geq S_H(f) + S_H((1/\tau T) - f)$ for $\tau < 1$, because $S_H(f)$ is bandlimited to W Hz. Hence the conditions in Theorem 3 are satisfied and the corollary is proved. □

Finally, we state a sufficient condition for behavior 3, that is, $C(P_0,\tau T) < C_N(P_0)$ for some P_0 and $1/(2WT) \leq \tau \leq 1$.

Theorem 4. *Assume that*

$$\int_0^{1/2\tau T} S^2_{H,\mathrm{fo}}(f,\tau T)\mathrm{d}f > \frac{T}{2}.$$

Then there exists a $P > 0$ such that $C(P_0,\tau T) < C_N(P_0)$.

Proof. Start by Taylor-expanding $g(\xi)$ introduced in the proof of Theorem 3. Taylor expansion is allowed for sufficiently small ξ values, since the integrand in $g(\xi)$ is analytic and converging with respect to ξ, and this also holds for the second term in $g(\xi)$:

$$\begin{aligned} g(\xi) &= \int_0^{1/2\tau T} (\xi S_{H,\mathrm{fo}}(f,\tau T) - \frac{1}{2}(\xi S_{H,\mathrm{fo}}(f,\tau T))^2 + O(\xi^3))\mathrm{d}f \\ &\quad - \frac{1}{2T}(\xi T - \frac{1}{2}(\xi T)^2 + O(\xi^3)) \\ &= \frac{\xi^2}{2}\left(\frac{T}{2} - \int_0^{1/2\tau T} S_{H,\mathrm{fo}}(f,\tau T)^2 \mathrm{d}f\right) + O(\xi^3) \end{aligned} \tag{1.51}$$

because $\int_0^{1/2\tau T} \xi S_{H,\mathrm{fo}}(f,\tau T)\mathrm{d}f = \xi/2$. $O(\xi)$ is such that $O(\xi^3)/\xi^3$ is bounded when $\xi \to 0$. From above, we conclude that if $\int_0^{1/2\tau T} S_{H,\mathrm{fo}}(f,\tau T)^2\mathrm{d}f > T/2$, then by choosing ξ sufficiently small we can make the last expression in (1.51) negative. This proves the theorem. □

Hence, to have $C(P_0,\tau T) \geq C_N(P_0)$ for all P_0, a necessary condition is $\int_0^{1/2\tau T} S_{H,\mathrm{fo}}^2(f,\tau T)\mathrm{d}f \leq T/2$ according to Theorem 4. Observe that pulses satisfying Theorem 3 cannot fulfill Theorem 4, since

$$\int_0^{1/2\tau T} S_{H,\mathrm{fo}}^2(f,\tau T)\mathrm{d}f \leq \int_0^{1/2\tau T} S_{H,\mathrm{fo}}(f,\tau T)\, T\mathrm{d}f = T/2.$$

Now, going back to the beginning of this section and setting $S_H(f) = S_{HG}(f)$, we see that all the derived theorems regarding the derivative of constrained capacity hold for this spectrum as well. The assumption of a flat channel was used only to compare the constrained capacity of orthogonal signaling with that of non-orthogonal signaling, obtained by succesively increasing the signaling rate. For frequency selective channels, we answer the question whether the constrained capacity is monotonically increasing with signaling rate or not.

1.4.2 Numerical Results

In this section, the results for Case I are applied to actual pulse spectra. The studied spectra are two triangular spectra as well as the frequently used root raised cosine pulse (rtRC). The two triangular ones are

$$S_{H^{\mathrm{tri1}}}(f) = \begin{cases} T - T^2 f, & 0 \leq f \leq \frac{1}{T} \\ 0, & f > \frac{1}{T} \end{cases} \tag{1.52}$$

and

$$S_{H^{\text{tri2}}}(f) = \begin{cases} T^2 f, & 0 \le f \le \frac{1}{T} \\ 0, & f > \frac{1}{T}. \end{cases} \tag{1.53}$$

Because these spectra are antisymmetric about $f = 1/2T$ it follows that both pulses are T-orthogonal. In both cases $W = 1/T$. What can be said about these two? Since both are unit energy and T-orthogonal it follows that they yield equal constrained capacities at $\tau = 1$. Moreover, at $\tau = 1/(2WT) = 1/2$, we have

$$S_{H^{\text{tri1}},\text{fo}}(f, T/2) = S_{H^{\text{tri1}}}(f) = S_{H^{\text{tri2}},\text{fo}}(1/T - f, T/2).$$

Inserting these folded shapes in (1.38) we obtain the same integral and it follows that both spectra yield the same constrained capacity for $\tau = 1/2$.

Although the two spectra are similar, it will be shown that for $1/2 < \tau < 1$ they have starkly different constrained capacity properties.

The folded spectrum of (1.52) is:

$$S_{H^{\text{tri1}},\text{fo}}(f, \tau T) = \begin{cases} T - T^2 f, & 0 \le f \le \frac{1}{\tau T} - \frac{1}{T} \\ 2T - \frac{T}{\tau}, & \frac{1}{\tau T} - \frac{1}{T} < f \le \frac{1}{2\tau T} \end{cases} \tag{1.54}$$

From (1.52), we see that the pulse spectrum is continuous in $[0, W]$. The spectrum is also decreasing in $[0, W]$ and the smallest value of (1.54) in the interval $[\frac{1}{\tau T} - W, 1/2\tau T]$ occurs for $f = 1/2\tau T$. Hence the conditions of Theorem 2 are fulfilled and it follows that the triangular pulse spectrum in (1.52) has monotonically increasing constrained capacity with decreasing τ. Computing the constrained capacity numerically, we get the curves in Figure 1.5. The constrained capacity indeed increases with decreasing τ and is larger than $C_N(P_0)$ for $\tau < 1$, hence, signaling faster is always beneficial for the decreasing triangular spectrum.

Inspecting (1.54), it is easy to conclude that the conditions in Theorem 3 are satisfied and it follows that $C(P_0, \tau T) - C_N(P_0)$ increases monotonically with P_0.

The folded spectrum of $S_{H^{\text{tri2}},\text{fo}}(f, \tau T)$ is

$$S_{H^{\text{tri2}},\text{fo}}(f, \tau T) = \begin{cases} T^2 f, & 0 \le f < \frac{1}{\tau T} - \frac{1}{T} \\ \frac{T}{\tau}, & \frac{1}{\tau T} - \frac{1}{T} \le f \le \frac{1}{2\tau T}. \end{cases} \tag{1.55}$$

We see that $S_{H^{\text{tri2}},\text{fo}}(f, \tau T)$ does not satisfy the conditions in Theorem 2 as it is not decreasing. Neither does it satisfy Theorem 3. However, it satisfies Theorem 4 for some τ values, which implies that this spectrum yields a constrained capacity which is smaller than $C_N(P_0)$ for some τ and P_0. Thus, the two spectra have significantly different behaviors: $S_{H^{\text{tri1}}}(f)$ has behavior 1 but

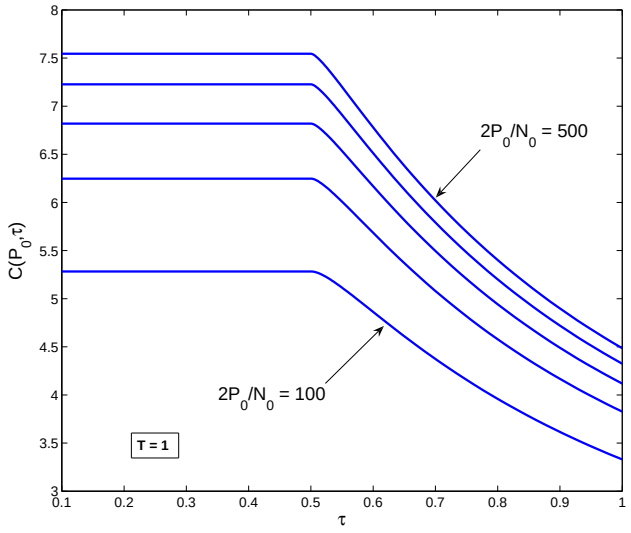

Figure 1.5: constrained capacity $C\left(P_0, \tau T\right)$ versus τ for $|H^{\text{tri1}}(f)|^2$.

$S_{H^{\text{tri2}}}(f)$ has behavior 3. In Figure 1.6 we have plotted $C\left(P_0, \tau T\right)$ versus τ for $S_{H^{\text{tri2}}}(f)$ for some values of P_0; it can be clearly seen that the constrained capacity for this pulse spectrum is indeed decreasing for some τ values.

Next we analyze the well known rtRC pulse. Its spectrum is given by [21]

$$S_H(f) = \begin{cases} T, & |f| \leq \frac{1-\beta}{2T} \\ T\cos^2\left(\frac{\pi T}{2\beta}\left(|f| - \frac{1-\beta}{2T}\right)\right), & \frac{1-\beta}{2T} < |f| \leq \frac{1+\beta}{2T} \\ 0, & |f| > \frac{1+\beta}{2T}, \end{cases} \tag{1.56}$$

where $0 \leq \beta \leq 1$. In this case, $W = (1+\beta)/2T$. The first conditions in Theorem 2 are trivially satisfied, since the rtRC spectrum is continuously differentiable and also has a decreasing spectrum. Next, we prove that the spectrum satisfies Corollary 1. Since only positive frequency values are studied, we can drop the magnitude operator sign in (1.56). The second derivative of (1.56) is

$$S_H''(f) = \begin{cases} 0, & f \leq \frac{1-\beta}{2T} \\ -\frac{\pi^2 T^3}{2\beta^2}\cos\left(\frac{\pi T}{\beta}\left(f - \frac{1-\beta}{2T}\right)\right), & \frac{1-\beta}{2T} < f \leq \frac{1+\beta}{2T} \\ 0, & f > \frac{1+\beta}{2T}. \end{cases} \tag{1.57}$$

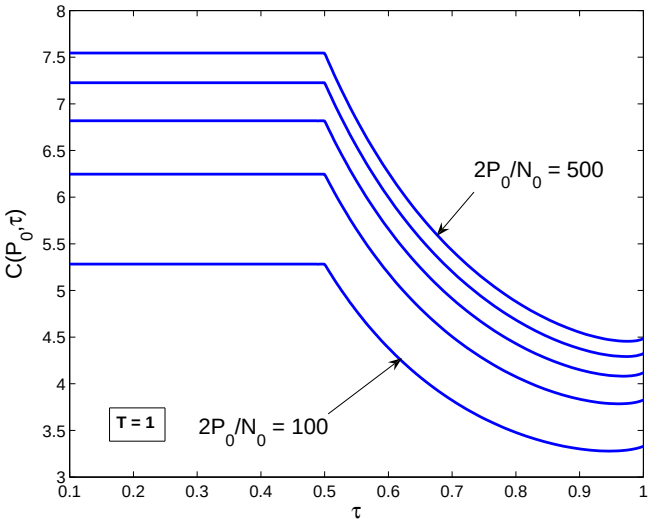

Figure 1.6: Constrained capacity $C\left(P_0, \tau T\right)$ versus τ for $|H^{\mathrm{tri2}}(f)|^2$.

From (1.57) it is seen that $S_H^{''}(f) \leq 0$ when $\frac{\pi T}{\beta}\left(f - \frac{1-\beta}{2T}\right) \leq \frac{\pi}{2}$. This reduces to $f \leq 1/2T$, which shows that Corollary 1 is satisfied; thus a rtRC has behavior 1 for any value of β. By inspection it can be seen that the rtRC satisfies Corollary 2 and hence $C\left(P_0, \tau T\right) - C_N(P_0)$ is monotonically increasing in P_0 for any τ.

Hence, the numerical results in this section confirm the theoretical analysis from Section 1.4.1. For the practical rtRC pulse, it is beneficial, from an constrained capacity perspective, to always reduce τ. Similarly, the theoretical analysis explains why the simple triangular spectras behave so differently, which is confirmed by the numerical results.

1.4.3 $S_H(f)$ free to choose for every τ

The assumptions imply that we first have to optimize (1.33) over $S_H(f)$, and then study its behavior with respect to τ. Thus, instead of (1.33) we should

study

$$\tilde{C}(P_0 S_{\boldsymbol{A}}(f,\tau T),\tau T) = \max_{S_H(f)} \int_0^{1/2\tau T} \log_2\left(1+\frac{2P_0\, S_{\boldsymbol{A}}(f,\tau T)\, S_{HG,\mathrm{fo}}(f,\tau T)}{N_0}\right)\mathrm{d}f$$

subject to

$$\int_0^W S_{\boldsymbol{A}}(f,\tau T) S_H(f)\mathrm{d}f = 1. \tag{1.58}$$

Next, we prove that when $S_G(f)$ is decreasing, $\tilde{C}(P_0 S_{\boldsymbol{A}}(f,\tau T),\tau T)$ in (1.58) is non-decreasing.

Theorem 5. *Assume that $S_G(f)$ is decreasing in $[0,W]$. Then*

$$\tilde{C}'_\tau(P_0 S_{\boldsymbol{A}}(f,\tau T),\tau T) \le 0.$$

Proof. Choose $S_H(f)$ such that $S_H(f) = 0, f \ge 1/2\tau T$, so that all its energy is placed within the frequency interval $[0, 1/2\tau T]$. This choice of $S_H(f)$ clearly maximizes the integrand of $\tilde{C}(P_0 S_{\boldsymbol{A}}(f,\tau T),\tau T)$. Now if τ decreases, we can still choose the previously defined pulse for this lower τ, so $\tilde{C}(P_0 S_{\boldsymbol{A}}(f,\tau T),\tau T)$ will at least have the value it had for the higher τ. □

As a practical example, we mention that digital subscriber lines (DSL) typically have low-pass frequency characteristics [22]. Indeed, most wired communication channels can approximately be characterized as low-pass filters.

1.4.4 $S_{\boldsymbol{A}}(f,\tau T)$ free, $S_H(f)$ fixed

This case is the toughest one from an analytical point of view. Note that in Section 1.4.3, the freedom to choose $S_H(f)$ makes it possible to concentrate all the energy of the PSD $S_{\boldsymbol{A}}(f,\tau T)S_H(f)$ to the interval where $S_G(f)$ is the largest. This is not possible to do if $S_H(f)$ is fixed, since $S_{\boldsymbol{A}}(f,\tau T)$ is periodic with a period of $1/\tau T$, and $f = 1/2\tau T$ is a point where it starts to repeat itself. Thus, we have to derive the derivative of $\hat{C}(P_0,\tau T)$ defined in (1.34). By choosing a sufficiently large P_0, $\theta(P_0,\tau T)$ becomes large and (1.34) reduces

to

$$\hat{C}(P_0,\tau T) = \int_0^{1/2\tau T} \log\left(\theta(P_0,\tau T)\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}\right)\mathrm{d}f$$
$$\text{subject to} \tag{1.59}$$
$$\int_0^{1/2\tau T} \theta(P_0,\tau T) - \frac{S_{H,\mathrm{fo}}(f,\tau T)}{S_{HG,\mathrm{fo}}(f,\tau T)}\mathrm{d}f = P_0/2.$$

We now have

Theorem 6. *For sufficiently large $P_0 < \infty$, $\hat{C}'_\tau(P_0,\tau T) < 0$ for $\tau \geq 1/(2WT)$.*

Proof. We start off, just as in the previous proofs, to split our integral into two parts.

$$\hat{C}(P_0,\tau T) = \int_0^{1/\tau T - W} \ln\left(\theta(P_0,\tau T)\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}\right)\mathrm{d}f + \int_{1/\tau T - W}^{1/2\tau T} \ln\left(\theta(P_0,\tau T)\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}\right)\mathrm{d}f. \tag{1.60}$$

We differentiate the expression in (1.60) with respect to τ, by using the Leibniz rule:

$$\hat{C}'_\tau(P_0,\tau T) = \int_0^{1/\tau T - W} \frac{\theta'_\tau(P_0,\tau T)}{\theta(P_0,\tau T)}\mathrm{d}f + \int_{1/\tau T - W}^{1/2\tau T} \frac{\theta'_\tau(P_0,\tau T)}{\theta(P_0,\tau T)} + \frac{\left(\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}\right)'_\tau}{\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}}\mathrm{d}f - \ln\left(\theta(P_0,\tau T)\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}\right)\frac{1}{2T\tau^2}. \tag{1.61}$$

By differentiating the second integral equation in (1.59) with respect to τ, we get the following expression

$$\int_0^{1/\tau T-W} \theta'_\tau(P_0,\tau T)\mathrm{d}f + \int_{1/\tau T-W}^{1/2\tau T} \left(\theta'_\tau(P_0,\tau T) - \left(\frac{S_{H,\mathrm{fo}}(f,\tau T)}{S_{HG,\mathrm{fo}}(f,\tau T)}\right)'_\tau\right)\mathrm{d}f - \frac{1}{2T\tau^2}\left(\theta(P_0,\tau T) - \frac{S_{H,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)}{S_{HG,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)}\right) = 0. \tag{1.62}$$

From (1.62) we get

$$\int_0^{1/2\tau T} \frac{\theta'_\tau(P_0,\tau T)}{\theta(P_0,\tau T)}\mathrm{d}f = \frac{1}{2T\tau^2}\left(1 - \frac{S_{H,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)}{S_{HG,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)\theta(P_0,\tau T)}\right) + \int_{1/\tau T-W}^{1/2\tau T} \left(\frac{S_{H,\mathrm{fo}}(f,\tau T)}{S_{HG,\mathrm{fo}}(f,\tau T)}\right)'_\tau \frac{1}{\theta(P_0,\tau T)}\mathrm{d}f. \tag{1.63}$$

Inserting (1.63) into (1.61), we get the following expression for $\hat{C}'_\tau(P_0,\tau T)$

$$\begin{aligned}\hat{C}'_\tau(P_0,\tau T) \;=\; & \int_{1/\tau T-W}^{1/2\tau T} \left(\left(\frac{S_{H,\mathrm{fo}}(f,\tau T)}{S_{HG,\mathrm{fo}}(f,\tau T)}\right)'_\tau \frac{1}{\theta(P_0,\tau T)} \right. \\ & + \frac{S_{H,\mathrm{fo}}(f,\tau T)}{S_{HG,\mathrm{fo}}(f,\tau T)}\left(\frac{S_{HG,\mathrm{fo}}(f,\tau T)}{S_{H,\mathrm{fo}}(f,\tau T)}\right)'_\tau\Bigg)\mathrm{d}f \\ & + \frac{1}{2T\tau^2}\left(1 - \frac{S_{H,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)}{S_{HG,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)\theta(P_0,\tau T)}\right. \\ & \left. - \ln\left(\theta(P_0,\tau T)\frac{S_{HG,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)}{S_{H,\mathrm{fo}}(\frac{1}{2\tau T},\tau T)}\right)\right).\end{aligned} \tag{1.64}$$

It is implicitly assumed that differentiation is permitted at all places where it occurs, and that none of the integrands are degenerate (division by 0). Since the integral in (1.64) is bounded, we see that by increasing P_0 and thus $\theta(P_0,\tau T)$, we will at some point have $\hat{C}'_\tau(P_0,\tau T) < 0$. This proves the theorem. □

Hence, Theorem 6 states that by just having a sufficiently large P_0, we can guarantee that the capacity will increase for increasing signaling rate, no matter the shape of the channel and the pulse. The threshold for the power P_0 depends on the channel and the pulse used, but is guaranteed to be a finite number. More clearly, Theorem 6 states the following fact: Even though there is frequency selection, which destroys the shape of the pulse, by performing waterfilling and signaling above a certain finite power level, increasing the signaling rate in linear modulation will increase the capacity.

1.5 Discussion

This work analyzed the behavior of the constrained capacity and capacity for linear modulation signaling when the signaling rate is changed. Both the cases of frequency selective and flat channels are investigated. Further, we allow for both fixed spectrum shaping filters as well as flexible ones, and in addition also data correlation. For flat channels, it is shown that there are pulses for which the constrained capacity can increase or *decrease* with signaling rate. Sufficient conditions on the pulse shape have been derived for both cases to occur. For frequency selective channels, it is shown that there is a threshold for the power which makes the capacity increasing when waterfilling is performed. As a general rule of thumb, it can be said that for channels and pulses that have approximately low-pass filter characteristics, it is in general beneficial to increase the signaling rate. Thus, for a given available computational complexity, the signaling rate should be as high as possible such that the complexity of the decoding does not overshoot the complexity constraint. In [34], an information rate analysis is performed that takes the decoding complexity constraint into account. Thus, a future extension of this work is to study the impact of the signaling rate on the achievable rates derived therein.

Part II

MIMO Precoding

Chapter 2

Introduction

2.1 Linear Communication Channels

A linear communication channel is characterized by the fact that the output signal without noise is a linear mapping of the input signal. A general mathematical model of linear channels in discrete time is

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}. \tag{2.1}$$

In (2.1), $\boldsymbol{y}$ is the $N_r \times 1$ received vector, $\boldsymbol{H}$ an $N_r \times N_t$ matrix that represents the linear channel and $\boldsymbol{x}$ the $N_t \times 1$ vector of transmitted data symbols. Throughout the thesis, Gaussian noise is assumed, hence $\boldsymbol{n}$ is an $N_r \times 1$ vector of Gaussian noise variables $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{R_N})$, where $\boldsymbol{R_N}$ is the correlation matrix $\mathbb{E}\{\boldsymbol{n}\boldsymbol{n}^*\} = \boldsymbol{R_N}$. If $\boldsymbol{R_N} \neq N_0\boldsymbol{I}_{N_r \times N_r}$, then the noise vector $\boldsymbol{n}$ is colored. In the next section, we look at different communication scenarios that can be described by (2.1).

2.1.1 Applications of linear channels

Despite its simplicity, many different communication systems can be represented by the linear model in (2.1). They merely differ in the structure of $\boldsymbol{H}$ and the noise correlation matrix $\boldsymbol{R_N}$. We will now present well-known communication systems that can be recast into (2.1).

ISI channels

As seen in Part I of the thesis, the single-carrier channel is a linear channel, since the output is a convolution (a linear operation) of the channel impulse re-

sponse and the input signal. By matched filtering and sampling of the received signal, we get a discrete model as in (1.23), which is again a linear channel. Assume that the ISI $\boldsymbol{z}$ is finite and of length $2M+1$, i.e, $\boldsymbol{z} = \{z_{-M}, \ldots, z_M\}$. With notation from Part I, assume that there are N transmitted symbols $\boldsymbol{a} = \{a_0, \ldots, a_{N-1}\}$. It is easily seen that (1.23), for certain sampling instances kT, $k = 0, 1, \ldots, N-1$, can be represented in the form

$$\boldsymbol{y} = \sqrt{P_0 T}\boldsymbol{Z}\boldsymbol{a} + \boldsymbol{\eta}, \tag{2.2}$$

where

$$\boldsymbol{Z} = \begin{pmatrix} z_0 & \cdots & z_M & & \\ z_1^* & z_0 & \cdots & z_M & \\ \vdots & & \ddots & & \\ z_M^* & z_{M-1}^* & & \cdots & z_M \\ & & & \ddots & \\ & & z_M^* & \cdots & z_0 \end{pmatrix} \tag{2.3}$$

and

$$\boldsymbol{a} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{pmatrix}. \tag{2.4}$$

In this case, we have $N_t = N_r = N$. Further, $\boldsymbol{\eta}$ is the colored Gaussian noise sequence in (1.23). For the ISI model, $\boldsymbol{R_N}$ is a Hermitian matrix where the element at position (i, j) is z_{i-j}.

If the model (2.2) is whitened, a causal ISI sequence $\boldsymbol{h} = \{h_0, \ldots, h_M\}$ is obtained [23]. Then, $\boldsymbol{y} = \sqrt{P_0 T}\boldsymbol{Z}\boldsymbol{a} + \boldsymbol{\eta}$ becomes $\boldsymbol{y} = \sqrt{P_0 T}\boldsymbol{H}\boldsymbol{a} + \boldsymbol{n}$, but where $\boldsymbol{H}$ now takes the form

$$\boldsymbol{H} = \begin{pmatrix} h_0 & & & & \\ \vdots & & & & \\ h_M & \cdots & h_0 & & \\ & & \ddots & & \\ & & h_M & \cdots & h_0 \\ & & & & \vdots \\ & & & & h_M \end{pmatrix} \tag{2.5}$$

and $\boldsymbol{n}$ is white Gaussian noise (WGN). Further, $N_t = N$ and $N_r = N + M$.

OFDM system

Assume a causal, whitened ISI sequence $\boldsymbol{h} = \{h_0, \ldots, h_M\}$. Construct the following vector of data symbols plus a cyclic prefix

$$\boldsymbol{x}_c = \begin{pmatrix} x_{N-M} \\ \vdots \\ x_{N-1} \\ x_0 \\ \vdots \\ x_{N-1} \end{pmatrix}. \tag{2.6}$$

Hence, a copy of the last M symbols $(x_{N-M}, \ldots, x_{N-1})$ is transmitted in the beginning; this goes under the name of *cyclic prefix* and was proposed in [33]. The symbols $(x_0, \ldots, x_{N-1})$ are not data symbols, but precoded symbols as will be explained shortly. $\boldsymbol{x}_c$ is transmitted across the channel $\boldsymbol{H}$ in (2.5), and after removing the cyclic prefix at the receiver, the model in (2.2) can be written as

$$\boldsymbol{y} = \boldsymbol{H}_c\boldsymbol{x} + \boldsymbol{n}, \tag{2.7}$$

where $\boldsymbol{x} = (x_0, x_1, \ldots x_{N-1})^{\mathrm{T}}$ and $^{\mathrm{T}}$ stands for transpose. $\boldsymbol{H}_c$ is now a cyclic Toeplitz matrix due to the cyclic prefix, and therefore posseses an eigenvalue factorization $\boldsymbol{H}_c = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^*$, where $\boldsymbol{U}$ is a discrete Fourier Transform (DFT) matrix and $\boldsymbol{D}$ a diagonal matrix of eigenvalues [31]. The eigenvalues equal the Fourier transform of the ISI sequence $\boldsymbol{h}$ evaluated at the different frequencies

$$d_{k,k} = \sum_{j=0}^{M} h_j e^{-2\pi\sqrt{-1}\,kj/N}.$$

Hence, if one encodes $\boldsymbol{x}$ as $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{a}$, and the receiver constructs $\hat{\boldsymbol{y}} = \boldsymbol{U}^*\boldsymbol{y}$, the equivalent model becomes

$$\hat{\boldsymbol{y}} = \boldsymbol{D}\boldsymbol{a} + \hat{\boldsymbol{n}}, \tag{2.8}$$

where $\hat{\boldsymbol{n}} = \boldsymbol{U}^*\boldsymbol{n}$. Note that $\hat{\boldsymbol{n}}$ is also WGN since $\boldsymbol{U}$ is unitary. Thus, the original ISI channel is decoupled into a set of parallel channels and ISI is avoided. This technique is known as orthogonal frequency division multiplexing (OFDM) and was invented in [32]. The penalty for combating ISI is the repetition of M symbols, which is a spectral efficiency and a power loss; however, this loss can be made small if the data block N is long in comparison with M. Another drawback with OFDM is that the elements x_j in the symbol vector $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{a}$ tend to fluctuate much more than the data symbols a_j, resulting in a high peak-to-average power ratio.

Single user MIMO system

For ISI channels, the model in (2.1) represented a time-sampled sequence, i.e., the y_k are sampled symbols of the received signal at different time instances. Instead, if multiple antennas are used at the transmiter and the receiver, $\boldsymbol{y}$ is the received signal across the receving antenna array during *one* channel use. Hence, multiple transmit and receive antennas provide additional degrees of freedom in one time slot. The term used to denote this scenario is multiple input multiple output systems (MIMO), and was introduced in the works [36, 35, 37] and further analyzed in, e.g., [38, 39]. The channel entry $h_{i,j}$ at position (i, j) in $\boldsymbol{H}$ now represents the channel impulse response between receiver antenna i and transmitter antenna j, which in this case is assumed to not have ISI. A common model [24], although not always realistic, is that the entries $h_{i,j}$ are i.i.d. and $h_{i,j} \sim \mathcal{CN}(0, \sigma^2)$. The noise sequence $\boldsymbol{n}$ is a WGN sequence in MIMO channels, i.e., $\boldsymbol{R_N} = N_0 \boldsymbol{I}_{N_r \times N_r}$. The symbol vector $\boldsymbol{x}$, that is transmitted across the N_t transmitter antennas, is typically constrained to an average energy constraint

$$\mathbb{E}\{\boldsymbol{x}^* \boldsymbol{x}\} \leq P_0. \tag{2.9}$$

At the receiver terminal, the vector $\boldsymbol{y}$ is observed across the receive antenna array. This vector can now be jointly processed at the receiver in order to retrieve the transmitted vector $\boldsymbol{x}$. Section 2.3 describes common processing techniques for Single User MIMO channels.

Single user MIMO-OFDM systems

If the channel between transmitter antenna i and receiver antenna j is frequency selective, one can apply the cyclic prefix technique together with a DFT transform to decouple the frequency selective MIMO channel [24]. Each data stream $\boldsymbol{a}_i = \{a_{i,1}, a_{i,2}, \ldots\}$, $i = 1 \ldots N_t$, that is transmitted from antenna i is subject to a cyclic prefix and a DFT transform. After stripping off the cyclic prefix at the receiver, the channel on each of the N_c frequency components becomes

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{a}_{1:N_t,k} + \boldsymbol{n}_k, \quad k = 1, \ldots, N_c, \tag{2.10}$$

where $\boldsymbol{a}_{1:N_t,k}$ is the transmitted vector on carrier k. Hence, the frequency selective MIMO channel is decomposed into N_c flat MIMO channels $\boldsymbol{H}_k$, $k = 1, \ldots, N_c$, each a $N_r \times N_t$ matrix. The system of matrix equations in (2.10) can be represented as one matrix equation,

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{N_c} \end{pmatrix} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_2 & \boldsymbol{0} & \ldots \\ \vdots & & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{H}_{N_c} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{N_c} \end{pmatrix} + \begin{pmatrix} \boldsymbol{n}_1 \\ \vdots \\ \boldsymbol{n}_{N_c} \end{pmatrix}. \tag{2.11}$$

This is again the model in (2.1).

Multi user MIMO system

Assume a cellular system with K single antenna autonomous users that all should be served by a single base station with N_t antennas. In broadcast transmission (MIMO BC), the base station transmits a vector $\boldsymbol{x}$, and the received signal at terminal j is of the form

$$y_j = \boldsymbol{h}_j \boldsymbol{x} + n_j, \quad j = 1, \ldots, K, \tag{2.12}$$

where $\boldsymbol{h}_j$ is the $1 \times N_t$ channel between the base station and terminal j. Hence, the received signal at each terminal is of the form in (2.1), just as for the single user MIMO system, and is called a multiple input single output (MISO) system. The system of equations in (2.12) can be represented as (2.1), where $\boldsymbol{H}$ is a matrix with $\{\boldsymbol{h}_j\}$, $j = 1, \ldots, K$ as rows. The symbol vector $\boldsymbol{x}$ is a precoded version of a data vector $\boldsymbol{a}$, where a_k is a symbol intended to user k. Essentially, the precoding is such that tranmission to user k lies in the null space of the other users $j \neq k$, so that the symbol a_k does not cause any interference to the users $j \neq k$. Thus, it is critical to provide the base station with accurate channel state information (CSI). Common precoding techniques are dirty paper coding (DPC) [28], zero forcing (ZF) [25], block diagonalization (BD) [26], and vector perturbation (VP) [27]. Note that, since the receiving terminals do not cooperate, joint detection of the vector $\boldsymbol{y}$ in (2.1) is not possible.

In a multiple access scenario (MIMO-MAC), each terminal transmits a symbol x_j to the base station. Let $\boldsymbol{h}_j^*$ be the channel between terminal j and the base station. The received signal at the base station is

$$\boldsymbol{y} = \boldsymbol{H}^* \boldsymbol{x} + \boldsymbol{n}, \tag{2.13}$$

where $\boldsymbol{H}^*$ is $N_t \times K$ and has $\{\boldsymbol{h}_j^*\}$ as columns. Hence, the MIMO MAC system model can also be represented with (2.1). When the base station decodes a symbol x_j from user j, it faces interference from the other user symbols.

2.1.2 Channel state information

When it comes to ISI channels, it was assumed that the ISI sequence $\boldsymbol{z}$ is perfectly known at the receiver and the transmitter, i.e., the channel matrix $\boldsymbol{H}$ in (2.1) is perfectly known. This assumption is reasonable for the case of deterministic channels, since $\boldsymbol{z}$ is then completely determined from the pulse $h(t)$, which is assumed to be known to the transmitter and the receiver. In the case of quasi-static frequency selective channels $g(t)$, a good enough channel

estimate can be obtained. For example, in digital subscriber lines (DSL), it is possible to obtain a good estimate of the channel [22].

However, for MIMO channels, the channel depends on the environment, which determines how fast it changes from one channel use to another. As soon as the channel variations are small, characterized by the coherence time T_C and the coherence bandwidth B_C, it is possible to track the channel and obtain reliable CSI. To obtain CSI at the receiver, a popular method is to send known pilot symbols from the transmitter to the receiver [29]. Decoding them at the receiver makes it possible to obtain an estimate $\hat{\boldsymbol{H}}$ of the channel $\boldsymbol{H}$, which can be made accurate by devoting more resources to the training phase. The amount of pilot data that needs to be transmitted has been analyzed in [29]. Note that transmitting pilot symbols, estimating the channel based on the pilot observations and using the estimate as if it is correct in the subsequent data detection phase, is not the optimal approach as it is inferior to performing a non-coherent detection. The ultimate limits of non-coherent detection have been studied in [30]. However, pilot tranmission yields significantly less computational complexity at the receiver side.

In order for the transmitter to obtain $\boldsymbol{H}$, two common techniques are:

1. **Feedback**: In this approach, the estimated channel $\hat{\boldsymbol{H}}$ is sent from the receiver to the transmitter on a feedback link. This feedback inherently gives rise to some delay δ. In order for $\hat{\boldsymbol{H}}$ to be reliable at the transmitter, we must have $\delta \ll T_C$. If the channel varies rapidly, this approach requires more frequent estimates $\hat{\boldsymbol{H}}$ and feedback.

2. **Channel Reciprocity**: This technique uses the assumption that the estimated channel from the transmitter to the receiver is the same as the channel from the receiver to the transmitter. Problems with this technique include calibration issues as well as the fact that the forward and backward channels are not necessarily close in time and frequency [24].

Despite the practical difficulties in obtaining perfect CSI, in situations when the channel varies slowly and the feedback link has sufficient capacity, the perfect CSI assumption is assumed to hold throughout this thesis.

2.2 Performance Measures For MIMO Channels

From now on, the focus of Part II is to analyze the spatial single user MIMO channel. The analysis is also applicable to general linear channels, but the used

notation is from MIMO communications. We start by describing two widely used performance measures for MIMO channels.

2.2.1 Information rate

The ultimate performance measure for a MIMO system is governed by its mutual information, which determines the achievable information rates for the MIMO channel. As described in Section 2.1.1, during one channel use, it is possible to transmit information along the spatial dimensions. The spatial channel during one such slot is described by the matrix $\boldsymbol{H}$. Assume that the channel does not change from one slot to another (quasi-static channel). Let $p_{\boldsymbol{X}}(\boldsymbol{x})$ denote the joint pdf/pmf of the vector $\boldsymbol{X} = [X_1, \ldots, X_{N_t}]^{\mathrm{T}}$ of N_t random variables[1]. It is assumed that $\boldsymbol{X}$ has zero mean $\mathbb{E}\{\boldsymbol{X}\} = \boldsymbol{0}_{N_t}$ and covariance matrix $\boldsymbol{R}_{\boldsymbol{X}} = \mathbb{E}\{\boldsymbol{X}\boldsymbol{X}^*\}$. The mutual information $\mathcal{I}(\boldsymbol{Y};\boldsymbol{X})$ between the input $\boldsymbol{x}$ and output $\boldsymbol{y}$ for a MIMO channel is defined as

$$\mathcal{I}(\boldsymbol{Y};\boldsymbol{X}) \triangleq \mathcal{H}(\boldsymbol{Y}) - \mathcal{H}(\boldsymbol{Y}|\boldsymbol{X}), \tag{2.14}$$

where $\mathcal{H}(\cdot)$ is the differential entropy operator [41]

$$\mathcal{H}(\boldsymbol{Y}) = -\int_{\boldsymbol{y}} p_{\boldsymbol{Y}}(\boldsymbol{y}) \log_2(p_{\boldsymbol{Y}}(\boldsymbol{y}))\, \mathrm{d}\boldsymbol{y}.$$

$\mathcal{I}(\boldsymbol{Y};\boldsymbol{X})$ is the number of bits that can be carried by $\boldsymbol{X}$ through $\boldsymbol{H}$, given the specified pdf $p_{\boldsymbol{X}}(\boldsymbol{x})$.

Definition 4. *The* information rate $I(\boldsymbol{H}, p_{\boldsymbol{X}}) = \mathcal{I}(\boldsymbol{Y};\boldsymbol{X})$ *is the maximum number of bits per channel use that can be carried error free through the MIMO channel* $\boldsymbol{H}$*, given the pdf* $p_{\boldsymbol{X}}(\boldsymbol{x})$*.*

If one maximizes $I(\boldsymbol{H}, p_{\boldsymbol{X}})$ over the pdf $p_{\boldsymbol{X}}(\boldsymbol{x})$, but keeps the correlation matrix $\boldsymbol{R}_{\boldsymbol{X}}$ fixed, one obtains the *constrained capacity* for the MIMO channel.

Definition 5. *The* constrained capacity *for a MIMO channel is*

$$C(\boldsymbol{H}, \boldsymbol{R}_{\boldsymbol{X}}) = \sup_{p_{\boldsymbol{X}}(\boldsymbol{x}):\mathbb{E}\{\boldsymbol{X}\boldsymbol{X}^*\}=\boldsymbol{R}_{\boldsymbol{X}}} I(\boldsymbol{H}, p_{\boldsymbol{X}}).$$

Soon we will present a closed form expression for the constrained capacity of the MIMO channel. Finally, maximizing $C(\boldsymbol{H}, \boldsymbol{R}_{\boldsymbol{X}})$ over $\boldsymbol{R}_{\boldsymbol{X}}$, yields the *capacity* for a MIMO channel. This maximization is valid only if there is a constraint on $\boldsymbol{R}_{\boldsymbol{X}}$, and the average power constraint is commonly used.

[1] A capital and bold symbol denote a vector of random variables, except for matrices which always are in capital and bold.

Definition 6. *The* capacity *for a MIMO channel is*

$$\hat{C}(P_0, \boldsymbol{H}) = \max_{\boldsymbol{R_X}:\mathrm{tr}(\boldsymbol{R_X})\leq P_0} C(\boldsymbol{H}, \boldsymbol{R_X}).$$

Note that the information rate per channel use is given by the mutual information between two sequences of random variables, $\boldsymbol{Y}$ and $\boldsymbol{X}$. Comparing this to Definition 1 in Part I of the thesis, we see that information rate for single antenna systems is also the mutual information of two sequences, but divided by their length (the number of channel uses). Thus, it can be expected that the information rate for MIMO systems is superior to single antenna systems (since no division by the length occurs), and this turns out to be the case.

Telatar derived exact analytical expressions for the constrained capacity $C(\boldsymbol{H}, \boldsymbol{R_X})$ and the capacity $\hat{C}(P_0, \boldsymbol{H})$ [39] of a MIMO system. The constrained capacity of a MIMO system is given by

$$C(\boldsymbol{H}, \boldsymbol{R_X}) \triangleq \max_{p_{\boldsymbol{X}}(\boldsymbol{x}):\mathbb{E}\{\boldsymbol{X}\boldsymbol{X}^*\}=\boldsymbol{R_X}} I(\boldsymbol{H}, p_{\boldsymbol{X}}) = \log_2 \det\left(\boldsymbol{I}_{N_t} + \frac{1}{N_0}\boldsymbol{H}\boldsymbol{R_X}\boldsymbol{H}^*\right). \tag{2.15}$$

The constrained capacity in (2.15) is attained by a multivariate Gaussian distribution on $\boldsymbol{X}$, with the correlation matrix $\boldsymbol{R_X}$. The capacity is obtained by subsequent maximization of (2.15) as in Definition 6. The solution to this optimization is the well-known waterfilling technique. Let $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*$ be the SVD decomposition of the channel $\boldsymbol{H}$ and $\boldsymbol{R_X} = \boldsymbol{Q}\boldsymbol{\Sigma}\boldsymbol{Q}^{\mathrm{T}}$ be the eigenvalue decomposition of $\boldsymbol{R_X}$. Further, let $\sigma_{j,j}$ be the diagonal elements of $\boldsymbol{\Sigma}$ and $s_{j,j}$ the diagonal elements of $\boldsymbol{S}$, respectively. The optimization in Definition 6 can be shown to be equivalent to

$$\hat{C}(P_0, \boldsymbol{H}) = \max_{\sum_{j=1}^{r}\sigma_{j,j}=P_0} \sum_{k=1}^{r} \log_2\left(1 + \frac{\sigma_{j,j}s_j^2}{N_0}\right), \tag{2.16}$$

where r is the rank of the channel $\boldsymbol{H}$. The unitary matrix $\boldsymbol{Q}$ is equal to $\boldsymbol{V}$. The solution to (2.16) is

$$\sigma_{j,j}^{\mathrm{opt}} = \left(\mu - \frac{N_0}{s_{j,j}}\right)_+, \tag{2.17}$$

where

$$x_+ = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

for a number x. Hence, in order to achieve the rate in (2.16), the transmitted vector $\boldsymbol{x}$ is constructed as $\boldsymbol{x} = \boldsymbol{V}\sqrt{\boldsymbol{\Sigma}}\boldsymbol{a}$, where $\boldsymbol{a}$ is a zero mean circularly

symmetric complex Gaussian (ZMCSCG) with $\boldsymbol{R_A} = \boldsymbol{I}_{N_t \times N_t}$. This transforms the linear channel in (2.1) into a set of parallel channels,

$$y_k = s_{k,k}\sigma_{k,k}^{\text{opt}} a_k + n_k, \quad k = 1, \ldots, r. \tag{2.18}$$

Thus, optimal transmission over the linear channel occurs over its eigenmodes $\{s_{j,j}\}$. Note, however, that this is only true if the data can be a multivariate Gaussian. As soon as the symbols x_j in $\boldsymbol{x}$ are drawn from a discrete constellation, which is the main focus of Part II in the thesis, signaling as in (2.18) is not optimal!

Transmitting at bit rates in (2.15) and (2.16), the probability of detecting an erroneous message at the receiver can be made arbitrarily close to 0 with long data blocks, in theory. Assume that the transmitter wants to convey 2^k different messages to the receiver. A bit pattern $\boldsymbol{b}$ of k bits is used to represent each message. This bit pattern is represented by a sequence of vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ which are sent through $\boldsymbol{H}$ in n different channel uses. The rate R of the system is defined as $R \triangleq k/n$ bits / channel use. Assuming that channel is used indefinitely, i.e., $n \to \infty$, it is possible to recover the transmitted message with error probability tending to zero as long as $R < C(\boldsymbol{H}, \boldsymbol{R_X})$ or $\hat{C}(P_0, \boldsymbol{H})$, if the encoding of the 2^k messages to the sequence of n vectors is done in an optimal fashion.

However, signaling exactly at these rates in practice is impossible for several reasons. First of all, it requires that the transmitted symbols x_j are taken from a Gaussian alphabet, which is not very practical. Moreover, the number of messages, k, has to be infinite (in theory), i.e, the transmission has to occur for an indefinite amount of time. Infinitely many vectors $\boldsymbol{x}_i$, $i = 1, \ldots$, need to be transmitted, and the receiver has to receive the whole signal $\boldsymbol{y}_i$, $i = 1, \ldots$, in order to make optimal detection. Still, it is possible to come close to these rates by modern coding systems. A popular method [42] is to code the bit stream $\boldsymbol{b}$ with, e.g., a low density parity check (LDPC) code, into a new bit sequence $\boldsymbol{c}$ of length $m > k$. Thus, the rate of the encoder is $R_c = k/m$. Next, the bits in $\boldsymbol{c}$ are mapped onto a discrete alphabet $\mathcal{X}$ (e.g. QAM) of cardinality $|\mathcal{X}| = M$. This creates a sequence of $n = m/N_t \log_2(M)$ symbol vectors $\boldsymbol{x}_j^{\text{T}} = [x_{j,1}, \ldots, x_{j,N_t}]$, $j = 1, \ldots, n$. The sequence of vectors is passed through a serial to parallel converter and then transmitted from the antenna array. Hence, there are $n = m/N_t \log_2(M)$ channel uses, and the total rate of the system is $R = R_c \log_2(M) N_t$ bits per channel use. This transmitter is shown in Figure 2.1. At the receiver, an iterative decoding algorithm is applied, that iterates between decoding the MIMO channel and the LDPC encoder, in accordance with the Turbo principle [40].

As soon as the alphabet for the symbols x_j is constrained to be discrete,

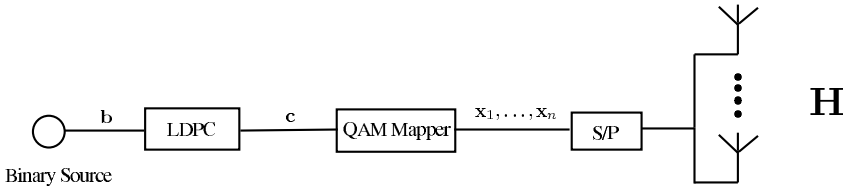

Figure 2.1: A practical transmission systems that can come close to the rates in Definitions 4 - 6. The bit sequence $\boldsymbol{b}$ is encoded into a much longer bit sequence $\boldsymbol{c}$ by an LDPC encoder. The bits in $\boldsymbol{c}$ are mapped onto a QAM constellation, which results in a sequence of vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, each with N_t symbols. After the serial to parallel converter S/P, each vector is transmitted from the antenna array across the channel $\boldsymbol{H}$.

the rates in (2.15) and (2.16) can never be reached exactly. Instead, the limit is $I(\boldsymbol{H}, p_{\boldsymbol{X}})$ in Definition 4. However, by using large QAM alphabets, $I(\boldsymbol{H}, p_{\boldsymbol{X}}) \approx C(\boldsymbol{H}, \boldsymbol{R}_{\boldsymbol{X}})$. Beside a large QAM alphabet, long codewords need to be produced by the encoder in order to reach $I(\boldsymbol{H}, p_{\boldsymbol{X}})$ and thereby $C(\boldsymbol{H}, \boldsymbol{R}_{\boldsymbol{X}})$. For an LDPC encoder, the needed block lengths can be as large as 10^5 [42, 43, 44]. For some time-critical applications, it is instead of interest to send short codewords and have less latency at the receiver side. This will inevitably lead to an error probability that is bounded away from 0. Furthermore, the alphabet $\mathcal{X}$ is in practice discrete. For these reasons, it is of interest to also consider other performance measures than information rate.

2.2.2 Bit and block error rate

As discussed in the previous section, once the codewords are relatively short, it will not be possible to signal at a vanishing error rate. It is then of interest to quantify the bit error rate (BER), the ratio of erroneous bits in the decoding of the message to the total number of bits, or the block error rate (BLER), the ratio of erroneous bit sequences to the total number of sequences. Clearly, these quantities should be as small as possible. Let $\boldsymbol{b}$ be the sequence of k bits to be sent across the linear channel. As in Figure 2.1, these bits are in general sent through an encoder that produces a longer sequence of bits, $\boldsymbol{c}$, that are mapped onto the discrete alphabet $\mathcal{X}$. This produces a sequence of vectors $\boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ that are transmitted through $\boldsymbol{H}$. The receiver observes the sequence $\boldsymbol{y}_{1:n} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ and produces an esimate $\hat{\boldsymbol{b}}$ of the sent bit sequence $\boldsymbol{b}$. Let b_i denote the ith bit in $\boldsymbol{b}$ and $P_i \triangleq \Pr\{\hat{b}_i \neq b_i\}$ be its error

probability. We can now define the bit error probability P_e as

$$P_e \triangleq \frac{\sum_{i=1}^{k} P_i}{k}. \tag{2.19}$$

Further, we define the block error probability P_e^k as

$$P_e^k \triangleq \Pr\{\hat{\boldsymbol{b}} \neq \boldsymbol{b}\}. \tag{2.20}$$

From the definition of P_e^k, it holds that

$$P_e^k = \Pr\{\cup\{\hat{b}_i \neq b_i, 1 \leq i \leq k\}\}. \tag{2.21}$$

Using the union bound, we get

$$P_e^k = \Pr\{\cup\{\hat{b}_i \neq b_i, 1 \leq i \leq k\}\} \leq \sum_{j=1}^{k} \Pr\{\hat{b}_i \neq b_i\} = kP_e. \tag{2.22}$$

Moreover, from the expression in (2.21), it is clear that

$$P_e^k \geq \Pr\{\hat{b}_i \neq b_i\}, \quad i = 1, \ldots, k. \tag{2.23}$$

Hence,

$$\sum_{i=1}^{k} P_e^k \geq \sum_{i=1}^{k} P_i,$$

which gives

$$P_e^k \geq P_e. \tag{2.24}$$

Combining (2.22) and (2.24), we get

$$P_e \leq P_e^k \leq kP_e. \tag{2.25}$$

Hence, minimizing the block error probability has a direct impact on the bit error probability and vice versa.

2.3 Receiver Structures for MIMO Channels

The information rates in Section 2.2.1 can be achieved by using the optimal decoder. However, if another decoder or detection method is used, then the performance degrades [47, 48]. For suboptimal detection, the ultimate limits, under certain conditions, can be derived through the framework of generalized mutual information [47, 48]. A thorough review of common detection methods for MIMO channels, both optimal and suboptimal, is given in [45]. We start by describing the optimal receiver first, followed by suboptimal techniques of lower complexity.

2.3.1 ML receiver

Let $\Pr\{\hat{\boldsymbol{b}} = \boldsymbol{b}\} = 1 - \Pr\{\hat{\boldsymbol{b}} \neq \boldsymbol{b}\} = 1 - P_e^k$ be the probability of a correct decision of the transmitted bit sequence at the receiver. Further, let $p_{\boldsymbol{Y}_{1:n}}(\boldsymbol{y}_{1:n})$ be the pdf of the received sequence $\boldsymbol{y}_{1:n}$. Then, the probability that the decision $\hat{\boldsymbol{b}}$ is correct can be expressed as

$$\Pr\{\hat{\boldsymbol{b}} = \boldsymbol{b}\} = \int_{\boldsymbol{y}_1} \cdots \int_{\boldsymbol{y}_n} \Pr\{\hat{\boldsymbol{b}} \,\text{sent}|\boldsymbol{y}_{1:n}\} p_{\boldsymbol{Y}_{1:n}}(\boldsymbol{y}_{1:n}) \prod_{k=1}^{n} \mathrm{d}\boldsymbol{y}_k. \tag{2.26}$$

Hence, $\Pr\{\hat{\boldsymbol{b}} = \boldsymbol{b}\}$ is maximized when the term $\Pr\{\hat{\boldsymbol{b}} \,\text{sent}|\boldsymbol{y}_{1:n}\}$ is maximized for every $\boldsymbol{y}_{1:n}$. Thus, given the received signal $\boldsymbol{y}_{1:n}$, the optimal decoding method is

$$\hat{\boldsymbol{b}} = \arg\max_{\tilde{\boldsymbol{b}}} \Pr\{\tilde{\boldsymbol{b}} \,\text{sent}|\boldsymbol{y}_{1:n}\}. \tag{2.27}$$

This is known as *maximum a posteriori* (MAP) decoding. This decoder minimizes the probability of detecting an erroneous message and also achieves the information rate. We can write

$$\Pr\{\tilde{\boldsymbol{b}} \,\text{sent}|\boldsymbol{y}_{1:n}\} = \frac{\Pr\{\boldsymbol{y}_{1:n}|\tilde{\boldsymbol{b}} \,\text{sent}\}\Pr\{\tilde{\boldsymbol{b}} \,\text{sent}\}}{p_{\boldsymbol{Y}_{1:n}}(\boldsymbol{y}_{1:n})}. \tag{2.28}$$

If all possible bit sequences $\boldsymbol{b}$ are equiprobable, it is readily seen from (2.28) that the maximization in (2.27) is equivalent to

$$\hat{\boldsymbol{b}} = \arg\max_{\tilde{\boldsymbol{b}}} \Pr\{\boldsymbol{y}_{1:n}|\tilde{\boldsymbol{b}} \,\text{sent}\}. \tag{2.29}$$

This is the *maximum likelihood* (ML) decoding rule, and the decoder is known as an *ML decoder*, which is thus optimal in the case of equiprobable bit sequences $\boldsymbol{b}$. In this thesis, we assume that $\boldsymbol{b}$ is indeed uniformly distributed. Clearly, this decoder minimizes the BLER.

Instead of performing ML decoding directly on the bit sequence $\boldsymbol{b}$, another method is to perform ML decoding on the transmitted sequence of vectors $\boldsymbol{x}_{1:n} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. First, let $\mathcal{V}$ denote the set of sequences $\boldsymbol{x}_{1:n}$ that are valid, i.e., each $\boldsymbol{x}_{1:n} \in \mathcal{V}$ represents a certain codeword that corresponds to some bit sequence $\boldsymbol{b}$. Then, the following ML decoding rule

$$\hat{\boldsymbol{x}}_{1:n} = \arg\max_{\tilde{\boldsymbol{x}}_{1:n} \in \mathcal{V}} \Pr\{\boldsymbol{y}_{1:n}|\tilde{\boldsymbol{x}}_{1:n} \,\text{sent}\} \tag{2.30}$$

is also optimal since each elements in the set $\mathcal{V}$ is equiprobable. Since (2.1) is an AWGN channel, (2.30) reduces to the following detection rule

$$\hat{\boldsymbol{x}}_{1:n} = \arg\min_{\boldsymbol{x}_{1:n} \in \mathcal{V}} \sum_{j=1}^{n} \|\boldsymbol{y}_j - \boldsymbol{H}\boldsymbol{x}_j\|^2. \tag{2.31}$$

In (2.31), $\|\cdot\|$ denotes the Frobenius norm $\|\boldsymbol{x}\|^2 = \mathrm{tr}(\boldsymbol{x}^*\boldsymbol{x})$ of a vector $\boldsymbol{x}$. Moreover, this norm is also well defined for a matrix argument, and will thus be used for matrices as well. Performing a joint decoding over the sequence $\boldsymbol{x}_{1:n}$ of transmitted vectors, which amounts to solving (2.31), implies a latency at the receiver. If an uncoded system is used, then vectors can be detected independently, which avoids the latency. We can then minimize each term in the summation in (2.31) independently, which gives rise to an ML decoding on the individual symbol vectors $\boldsymbol{x}_j$ at each time instant. Hence, we consider

$$\hat{\boldsymbol{x}}_k = \arg\min_{\tilde{\boldsymbol{x}}_k} \|\boldsymbol{y} - \boldsymbol{H}\tilde{\boldsymbol{x}}_k\|^2. \tag{2.32}$$

The probability of symbol vector error is

$$\Pr\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\} = 1 - \Pr\{\cup_{\tilde{\boldsymbol{x}}\neq\boldsymbol{x}} \|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}}) + \boldsymbol{n}\|^2 > \|\boldsymbol{n}\|^2\}. \tag{2.33}$$

Since $\Pr\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\}$ is hard to put in close form, we use the well known union bound to obtain an upper bound:

$$\begin{aligned}\Pr\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\} &\leq \Pr\{\boldsymbol{x}\,\text{sent}\} \sum_{\tilde{\boldsymbol{x}}\neq\boldsymbol{x}} \frac{1}{2^{|\mathcal{X}^N|}} \Pr\{\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}}) + \boldsymbol{n}\|^2 < \|\boldsymbol{n}\|^2\} \\ &= \Pr\{\boldsymbol{x}\,\text{sent}\} \sum_{\tilde{\boldsymbol{x}}\neq\boldsymbol{x}} \frac{1}{2^{|\mathcal{X}^N|}} \Pr\{\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}}) + \boldsymbol{n}\|^2 - \|\boldsymbol{n}\|^2 < 0\}\end{aligned} \tag{2.34}$$

In each term of the second summation in (2.34), the variables $\boldsymbol{H}$ and $\boldsymbol{x}$ are fixed. This implies that $\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}}) + \boldsymbol{n}\|^2 - \|\boldsymbol{n}\|^2 \sim \mathcal{N}(\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}})\|^2, 4\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}})\|^2)$. Hence,

$$\Pr\{\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}}) + \boldsymbol{n}\|^2 - \|\boldsymbol{n}\|^2 < 0\} = Q(\|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}})\|/2), \tag{2.35}$$

where Q is the Gaussian tail function

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2}\,\mathrm{d}y. \tag{2.36}$$

Since $Q(x)$ has a steep descent towards 0, the dominating terms in the summation in (2.34) are those corresponding to the minimum distance, i.e., $P\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\} \approx Q(\sqrt{D^2_{\min}(\boldsymbol{H}, \mathcal{X})})$ where

$$D^2_{\min}(\boldsymbol{H}, \mathcal{X}) = \min_{\boldsymbol{x}\neq\tilde{\boldsymbol{x}}} \|\boldsymbol{H}(\boldsymbol{x}-\tilde{\boldsymbol{x}})\|^2. \tag{2.37}$$

The minimum in (2.37) depends on the channel $\boldsymbol{H}$ and the alphabet $\mathcal{X}$. Thus, maximizing $D^2_{\min}(\boldsymbol{H}, \mathcal{X})$ has a strong impact on lowering the block error probability at high SNRs, which in turn lowers the BER. The minimum distance is the design parameter chosen in this thesis, and it will be analyzed for certain discrete alphabets $\mathcal{X}$.

2.3.2 Linear receivers

The decoding rule in (2.32) can be implemented as a tree search operating over $\boldsymbol{H}$. However, its complexity, i.e., the number of leaf nodes, is exponential in the alphabet size $|\mathcal{X}|$ and the memory of the channel $\boldsymbol{H}$ (spatial for MIMO or temporal for, e.g., ISI). Hence, to reduce the complexity of the detection, suboptimal and low complexity receivers are desirable. One low complexity receiver is a linear one, which outputs the estimate

$$\hat{\boldsymbol{x}} = T_{\mathcal{X}}(\boldsymbol{W}\boldsymbol{y}) \tag{2.38}$$

of the transmitted vector $\boldsymbol{x}$, where the operation $T_{\mathcal{X}}(\boldsymbol{r})$ of a vector $\boldsymbol{r}$ rounds each element r_j to the nearest element in $\mathcal{X}$. Depending on the performance measure, it may be the case that different $\boldsymbol{W}$ are optimal. However, it turns out that most performance measures of interest, such as information rate and bit error rate, are directly related to the mean-square-errors (MSEs) of a linear receiver [49]. The MSEs are the diagonal elements of the MSE matrix

$$\boldsymbol{E} \triangleq \mathbb{E}\{[\boldsymbol{W}\boldsymbol{y} - \boldsymbol{x}][\boldsymbol{W}\boldsymbol{y} - \boldsymbol{x}]^*\}. \tag{2.39}$$

The filter that minimizes the MSEs is optimal for these performance measures. This filter is the well-known *Wiener filter* [24], and equals

$$\boldsymbol{W} = (\boldsymbol{H}^*\boldsymbol{R}_{\boldsymbol{X}}\boldsymbol{H} + \boldsymbol{I})^{-1}\boldsymbol{H}^*. \tag{2.40}$$

It is readily seen that the MSEs are directly dependent on the correlation matrix $\boldsymbol{R}_{\boldsymbol{X}}$, which is determined by the discrete alphabet $\mathcal{X}^{N_t}$ of the vectors $\boldsymbol{x}$. Here, $\mathcal{X}^{N_t}$ is the N_t fold Cartesian product of the symbol alphabet $\mathcal{X}$.

2.4 Precoding for Linear Channels

Section 2.3 introduced different receiver structures, which produce different values for the introduced performance measures in Section 2.2. Furthermore, these receiver structures are directly dependent on the discrete alphabet $\mathcal{X}^{N_t}$. Thus, it is of interest to find the optimal discrete set $\mathcal{X}^{N_t}$ for the different receiver structures and performance measures. However, this is also a non-tractable problem if no constraints are put on the set $\mathcal{X}^{N_t}$, except for the obvious energy constraint. A widely used technique to generate $\mathcal{X}^{N_t}$, which starts to lend analytical tractability, is to construct the symbol vector $\boldsymbol{x}$ as

$$\boldsymbol{x} = \mathcal{L}(\boldsymbol{a}), \tag{2.41}$$

where $\boldsymbol{a}$ is a data vector that comes from a well-defined discrete alphabet $\mathcal{A}^B$, and $\mathcal{L}$ is a certain function/mapping. Note here that the mapping $\mathcal{L}$ is $\mathcal{L} : \mathbb{C}^B \to \mathbb{C}^{N_t}$, where $B \neq N_t$ can hold. In practice, the alphabet $\mathcal{A}$ is an M^2-QAM alphabet,

$$\mathcal{A} \triangleq \{z_r + iz_i \,:\, z_r, z_i \in \{(-M+1)/2, \ldots, (M-1)/2\}\}.$$

From now on, we will always let $\mathcal{A}$ be the QAM alphabet.

In general, to find the optimal mapping $\mathcal{L}$ for a certain receiver structure and performance measure is a very tough problem. The mappings can be divided into two classes: linear mappings and non linear mappings. The latter often give rise to higher complexity (either encoding or decoding complexity), but can in general perform better than linear mappings. However, linear mappings always have a linear encoding complexity, while the decoding complexity depends on the receiver, and is in general easy to estimate for linear mappings. We will now describe these two classes of mappings.

2.4.1 Linear precoding

When $\mathcal{L}$ is a linear map, it can be represented by a matrix equation, i.e., $\boldsymbol{x} = \boldsymbol{F}\boldsymbol{a}$ for some matrix $\boldsymbol{F}$. A linear map is called a *linear precoder*. Here, $\boldsymbol{a}$ is an $B \times 1$ vector, and $\boldsymbol{F}$ is $N_t \times B$. Since $\boldsymbol{a}$ is discrete and structured, so will $\boldsymbol{x}$ be. We have already seen an application of linear precoding. To achieve the capacity $\hat{C}(P_0, \boldsymbol{H})$ in (2.16), the vector $\boldsymbol{x}$ is constructed as $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{a}$, where $\boldsymbol{a} \sim \mathcal{CN}(\boldsymbol{0}_{N_t}, \boldsymbol{I}_{N_t \times N_t})$ (i.e., $B = N_t$). Hence, in this case, $\boldsymbol{F} = \boldsymbol{V}\boldsymbol{\Sigma}$. In general, since both the transmitter and receiver have perfect channel knowledge, it is possible to optimize over a linear transformation $\boldsymbol{F}$ of the data symbols $\boldsymbol{a}$, in order to improve a performance measure imposed on (2.1). Hence, a more general linear model than (2.1) arises from this consideration:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{F}\boldsymbol{a} + \boldsymbol{n}. \tag{2.42}$$

As in (2.1), $\boldsymbol{n} \sim \mathcal{CN}(\boldsymbol{0}_{N_r}, \boldsymbol{I}_{N_r \times N_r})$. Since $\boldsymbol{x} = \boldsymbol{F}\boldsymbol{a}$, $\boldsymbol{R}_{\boldsymbol{X}} = \mathbb{E}\{\boldsymbol{F}\boldsymbol{a}\boldsymbol{a}^*\boldsymbol{F}^*\} = \boldsymbol{F}\boldsymbol{R}_{\boldsymbol{A}}\boldsymbol{F}^*$, where the last equality follows from the linearity of $\mathbb{E}\{\cdot\}$ and the fact that $\boldsymbol{F}$ is not stochastic. Hence, if $\boldsymbol{R}_{\boldsymbol{A}} = \boldsymbol{I}_{B \times B}$, then $\boldsymbol{R}_{\boldsymbol{X}} = \boldsymbol{F}\boldsymbol{F}^*$. In this case, $\boldsymbol{F}$ determines the correlation matrix of $\boldsymbol{x}$. The constraint in (2.9) now becomes

$$\mathrm{tr}(\boldsymbol{F}\boldsymbol{R}_{\boldsymbol{A}}\boldsymbol{F}^*) \leq P_0. \tag{2.43}$$

Since $\boldsymbol{R}_{\boldsymbol{A}}$ is a correlation matrix, it is positive semidefinite, and therefore possesses a factorization of the form $\boldsymbol{R}_{\boldsymbol{A}} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ for some matrix $\boldsymbol{L}$. Hence, a new precoder can be defined, $\hat{\boldsymbol{F}} = \boldsymbol{F}\boldsymbol{L}$, which is subject to the constraint

$\mathrm{tr}(\hat{\boldsymbol{F}}\hat{\boldsymbol{F}}^*) \leq P_0$. Thus, without loss of generality (WLOG), we can assume that $\boldsymbol{R_A} = \boldsymbol{I}_{B\times B}$ in (2.43), since $\boldsymbol{L}$ can be incorporated into $\boldsymbol{F}$. Therefore, we consider uncorrelated QAM symbols.

2.4.2 Non-linear precoding

If $\mathcal{L}$ is a non-linear function, then we obtain a *non-linear precoder*. A common non-linear precoding technique is *vector perturbation* [51, 52, 55], already mentioned in Section 2.1.1, which perturbs the data vector $\boldsymbol{a}$ with another vector $\boldsymbol{p}$ that comes from a lattice, in order to reduce the transmit energy $\mathrm{tr}(\boldsymbol{F}\boldsymbol{R_A}\boldsymbol{F}^*)$. The data symbols a_j are assumed to belong to a bounded region in the complex-valued plane. Usually, this region is the cube $\mathcal{K} = \{a : |\mathrm{Re}\{a\}| < 0.5, |\mathrm{Im}\{a\}| < 0.5\}$, i.e., $a_j \in \mathcal{K}$ and $\boldsymbol{a} \in \mathcal{K}^B$, the B dimensional cube. A vector $\boldsymbol{s}$ is constructed as

$$\boldsymbol{s} = \boldsymbol{H}^+(\boldsymbol{a} + \boldsymbol{p}), \tag{2.44}$$

where $\boldsymbol{H}^+$ is the Moore-Penrose pseudo inverse of $\boldsymbol{H}$ and $\boldsymbol{p}$ is the solution to

$$\boldsymbol{p} = \arg \min_{\boldsymbol{q}\in\mathbb{Z}[i]^B} \|\boldsymbol{H}^+(\boldsymbol{a} + \boldsymbol{q})\|^2. \tag{2.45}$$

In (2.45), $\mathbb{Z}[i]^B$ denotes the set of B dimensional Gaussian integer vectors. The transmitted vector $\boldsymbol{x}$ is then

$$\boldsymbol{x} = \sqrt{\frac{P_0}{\kappa(\boldsymbol{H})}}\boldsymbol{s}, \tag{2.46}$$

where $\kappa(\boldsymbol{H})$ is the average energy

$$\kappa(\boldsymbol{H}) = \mathbb{E}_{\boldsymbol{A}}\{\|\boldsymbol{s}\|^2\} \tag{2.47}$$

with respect to the data vector $\boldsymbol{a}$. Hence, the received signal is

$$\boldsymbol{y} = \sqrt{\frac{P_0}{\kappa(\boldsymbol{H})}}\,(\boldsymbol{a} + \boldsymbol{p}) + \boldsymbol{n}. \tag{2.48}$$

Decoding $\boldsymbol{a}$ is now a simple matter. Since $\boldsymbol{p} \in \mathbb{Z}[i]^B$ and $\boldsymbol{a} \in \mathcal{K}^B$, the lattice vectors $\boldsymbol{p}$ translate the cube $\mathcal{K}^B$ so that it tiles the complex-valued B dimensional space $\mathbb{C}^B$; that is, the translated cubes cover $\mathbb{C}^B$ and they do not intersect. Let $\hat{a}_j$ denote the j:th decoded data symbol. Then

$$\hat{a}_j = y_j \bmod \mathcal{K}, \tag{2.49}$$

where $\mod \mathcal{K}$ means that y_j is translated to $\mathcal{K}$, i.e., $y_j - z_j \in \mathcal{K}$ for a (unique) z_j. Thus, a simple modulo operation for each received stream y_j recovers a_j. Note that the decoded $\hat{a}_j$ is corrupted by a modulo Gaussian noise, $n_k \mod \mathcal{K}$.

Vector perturbation gives rise to a simple decoding method, by inverting the channel and translating the data vector $\boldsymbol{a}$ to reduce the transmit energy. The main bottleneck is the computational complexity needed to find the optimal $\boldsymbol{p}$ in (2.45), which is a well-known NP-hard problem [52]. Hence, an NP-hard problem needs to be solved online for every realization of $\boldsymbol{H}$ and $\boldsymbol{a}$. A suboptimal low-complexity instance of vector perturbation is Tomlinson-Harashima precoding [53].

2.5 Construction of Linear Precoders

The low encoding complexity of linear precoders is very desirable for practical applications. For this reason, linear precoders have been an active area of research throughout the history of MIMO communications and is, e.g., incorporated in the long term evolution (LTE) standard [54]. As mentioned in the previous section, depending on the receiver structure and the different performance measures of interest, different optimal precoders are obtained. We will now review some linear precoding techniques for different receiver structures.

2.5.1 Optimal linear precoders for linear receivers

As described in Section 2.3.2, the Wiener filter is the optimal linear receiver for many performance measures of interest. Employing this filter at the receiver, the next task is to find linear precoders that maximize different performance measures. A thorough investigation of this problem is performed in [49, 50, 61]. The optimization problems that arise are efficiently solved with majorization techniques, and it turns out that the optimum precoder can be derived in a relatively easy fashion. Since now $\mathcal{L}(\boldsymbol{a}) = \boldsymbol{F}\boldsymbol{a}$, it holds that the MMSE matrix in (2.39) is $\boldsymbol{E} = (\boldsymbol{I}_{N_t,N_t} + \boldsymbol{F}^*\boldsymbol{H}^*\boldsymbol{H}\boldsymbol{F})^{-1}$. Let $N \leq \min(N_r, N_t)$. Given an arbitrary objective function $f(e_{1,1}, \ldots, e_{N,N})$ of the diagonal elements from $\boldsymbol{E}$ that is increasing in its arguments and minimized when its arguments are sorted in decreasing order, the solution to the optimization problem

$$\begin{aligned} &\min_{\boldsymbol{F}} f(\{e_{j,j}\}) \\ &\text{subject to} \\ &e_{j,j} \leq \rho_j, \quad j = 1, \ldots, N \\ &\operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^*) \leq P_0 \end{aligned} \tag{2.50}$$

is of the form $\boldsymbol{F} = \boldsymbol{V}\text{diag}(\sqrt{\boldsymbol{p}})\boldsymbol{Q}$. Here $\boldsymbol{V}$ is the right unitary matrix of $\boldsymbol{H}$, $\boldsymbol{Q}$ is a unitary matrix such that $e_{j,j} = \rho_j$, $j = 1, \ldots, N$, ($\boldsymbol{Q}$ can be obtained by a rather simple algorithm [49, Algorithm 2.2]) and $\text{diag}(\sqrt{\boldsymbol{p}})$ is a diagonal matrix with the vector $\sqrt{\boldsymbol{p}}$ on its main diagonal. As before, let $\boldsymbol{S}$ be the singular values of the channel $\boldsymbol{H}$. The vector $\boldsymbol{p}$ and the values $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]$ are obtained through the optimization

$$\begin{aligned} &\min_{\boldsymbol{\rho},\boldsymbol{p}} f(\boldsymbol{\rho}) \\ &\text{subject to} \\ &\sum_{j=i}^{N} \frac{1}{1 + p_j s_{j,j}} \leq \sum_{j=1}^{N} \rho_j, \quad \leq 1 \leq i \leq N \\ &\rho_i \geq \rho_{i+1} \\ &\sum_{j=1}^{N} p_j \leq P_0 \\ &p_j \geq 0, \quad 1 \leq j \leq N. \end{aligned} \tag{2.51}$$

It turns out that the BER function is convex in $\boldsymbol{\rho}$ as soon as it is below a certain threshold $\approx 10^{-3}$. Thus, the problem in (2.51) is a convex optimization problem, and minimizing the BER is a convex problem that can be solved efficiently with convex optimization techniques.

Instead, if the interest is to maximize the mutual information, the problem reduces to minimizing the determinant of $\boldsymbol{E}$ [49], and the optimal $\boldsymbol{F}$ has $\boldsymbol{Q} = \boldsymbol{I}_{N_t,N_t}$ and $p_i = (\mu - \lambda_{\boldsymbol{H},i}^{-1})_+$, where μ is such that $\sum_{j=1}^{N} p_j = P_0$ holds. From (2.51) it is thus possible to derive closed form solutions in the case of simple functions f, and also optimal numerical solutions when f is convex.

2.5.2 Optimal linear precoders for maximizing the mutual information

The Wiener filter is after all a suboptimal receiver, which thus gives suboptimal performance of a MIMO system. If instead the optimal ML decoding rule is employed at the receiver, the analysis of the optimal linear precoders is significantly tougher. Note that since $\boldsymbol{x} = \boldsymbol{F}\boldsymbol{a}$, $p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{A}}(\boldsymbol{a}) = 1/|\mathcal{A}^B|$ and $R_{\boldsymbol{X}} = \boldsymbol{F}\boldsymbol{F}^*$. The information rate $I(\boldsymbol{H}, p_{\boldsymbol{X}})$ in Definition 4 can be denoted as $I(\boldsymbol{H}, \boldsymbol{F})$, where the optimization variables are explicit. The information rate

optimal precoder is found by solving

$$\begin{aligned} &\max_{\boldsymbol{F}} I(\boldsymbol{H}, \boldsymbol{F}) \\ &\text{subject to} \\ &\mathrm{tr}(\boldsymbol{F}\boldsymbol{F}^*) \leq P_0. \end{aligned} \tag{2.52}$$

Note that it is trivial to solve (2.52) with $C(\boldsymbol{H}, \boldsymbol{R}_{\boldsymbol{X}})$ and $\hat{C}(P_0, \boldsymbol{H})$ as objective functions, i.e., when the alphabet $\mathcal{A}$ is Gaussian: The optimum linear precoder then performs waterfilling. However, since the alphabet $\mathcal{A}$ is discrete, there is no closed form expression available for the objective function $I(\boldsymbol{H}, \boldsymbol{F})$. Finding the precoder $\boldsymbol{F}$ that solves (2.52) is a challenging problem. In [83], the Karush-Kuhn-Tucker (KKT) conditions were derived for (2.52), which produced a fixed point equation for the optimal $\boldsymbol{F}$. Based on this equation, an iterative optimization technique was developed that produced precoders providing high information rate. However, the problem with this approach is that the iterative optimization technique is not guaranteed to converge to the optimum, since (2.52) is a non-convex problem in $\boldsymbol{F}$. A recent advance in [56] shows that $I(\boldsymbol{H}, \boldsymbol{F})$ is concave over the Gram matrix $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{H}^*\boldsymbol{H}\boldsymbol{F}$. This enables construction of an algorithm that converges to the optimum of (2.52). However, this algorithm is of very high complexity and is not very feasible for large QAM constellations and MIMO dimensions.

The work in [83] showed an interesting connection between information rate and the minimum distance $D^2_{\min}(\boldsymbol{H}\boldsymbol{F}, \mathcal{A})$. Namely, Theorem 4 in [83] shows that for high SNRs, the precoder solving (2.52) is the one maximizing the minimum distance $D^2_{\min}(\boldsymbol{H}\boldsymbol{F}, \mathcal{A})$. Thus, for high SNRs, the solution to (2.52) is obtained by optimizing the minimum distance. Hence, an interesting connection exists between three well-known optimization criterias: the minimum distance, BER and information rate. The precoder that minimizes the BER at high SNRs, maximizes the data rate and the minimum distance at the same time!

2.5.3 Linear precoders for minimizing the BER

In order to minimize the BER at *any* SNR, the common approach is to minimize the expression in (2.33). Since this expression cannot be put in closed form, different approximations of it are minimized, such as the Chernoff upper bound. In [57], a general expression for the precoder that minimizes (2.33) was presented. Similar results are derived in [58]. However, these expressions are given in terms of unknown matrices, and to determine these matrices is a non-tractable task in dimensions higher than two. Works such as [59, 60, 61, 62] present suboptimal constructions to minimize the BER. In general, finding the

precoder that minimizes (2.33) is a non-tractable problem, and approximations to (2.33) are made which relax the problem into a tractable one.

2.5.4 Linear precoders without CSI at transmitter

When the transmitter has no knowledge about the channel coefficients $\boldsymbol{H}$, the construction of precoders is of a different nature than before. In many cases, the receiver has the capability to obtain a good enough estimate of the channel. One alternative then is to feed back the CSI to the transmitter, as mentioned in Section (2.1.2), so that the transmitter has a channel to work with. However, due to the inherent delay, and in the case of low rate feedback links, this method is not viable. Instead, in this scenario, it is desirable that the receiver only feeds back a small amount of information to the transmitter, which is sufficient for determining the precoder at the transmitter. In [63], it was shown that for MIMO BC with a zero forcing precoder (ZF) at the base station, the number of feedback bits required increases linearly with the SNR.

Usually, the transmitter is equipped with an already static, finite collection of precoders, a *precoder codebook*, and the receiver only feeds back a string of bits across the MIMO channel, that represent the position of the precoder in the codebook that the transmitter should use. This is known as *limited feedback* precoding. Hence, the art of limited feedback precoding is in designing the finite precoder codebook. Many different techniques exist for this purpose. In [109], precoders with k orthogonal columns are designed, where $k < N_t$. It is shown that the optimal such precoder, for many performance measures, has its k columns isotropically (i.e., "evenly") distributed across the unitary space $\mathcal{U}(N_t, k)$ of $N_t \times k$ matrices with k orthogonal columns. Hence, the optimal codebook should consist of precoders that are evenly spread across $\mathcal{U}(N_t, k)$. By constructing different distance measures between the subspaces that each such precoder spans, it is possible to construct codebooks of different sizes containing evenly spread precoders. Beside orthogonal precoding, other methods exist that feedback a few bits representing different elements of a precoder that optimizes, e.g., the minimum distance [112].

In this thesis, we will make a comparison between different limited feedback schemes and improve upon previous ones for MMSE receivers. This is the subject of Chapter 6.

Chapter 3

Linear Precoders for Maximizing the Minimum Distance

Due to the difficulty in finding a precoder that minimizes the BER, a common method is to minimize a quantity directly related to the BER. As described in the previous chapter, the minimum distance is the dominant factor in the BER at high SNRs, and the precoder that maximizes it not only minimizes the BER, but also maximizes the information rate. There have been many attempts to construct precoders that increase the minimum distance, see, e.g., [58, 60, 61, 79] among many others. All of these attempt to produce suboptimal constructions for moderate dimensions of the MIMO system. In [67], the precoder that maximizes the minimum distance for two dimensional MIMO channels ($N_t = B = 2$ and $N_r \geq 2$) and 4-QAM alphabets was found. It is shown that there are essentially only two different precoder "structures" that are optimal. With "structure", it is meant that the mathematical expression for the precoder takes on two different forms, but the precoder itself changes continuously with the channel, c.f., [67]. As we will see later, this structure is characterized by the Gram matrix $\boldsymbol{G} = \boldsymbol{F}^* \boldsymbol{H}^* \boldsymbol{H} \boldsymbol{F}$: The two different precoder structures correspond to two different Gram matrices. For any MIMO channel $\boldsymbol{H}$ with a ratio of its singular values that is above a certain threshold, one of these structures is always optimal, while for a channel with a ratio below the threshold, the other structure is optimal. Thus, in a way, the optimal precoder behaves in a discrete fashion.

The goal of this chapter is to provide new insights into the design of linear

precoders $\boldsymbol{F}$ that maximize the minimum distance of the received signaling points. We start off with new suboptimal constructions in Section 3.2. Thereafter, we attempt to find the optimal minimum distance precoders through an iterative optimization technique, presented in Section 3.3. From the output of the optimization, we are able to observe a profound structure in the optimal solutions, which directs towards a possibility of an analytic treatment of the problem. This analysis is covered in Chapter 4.

3.1 Problem Under Consideration

The problem of finding the $N_t \times B$ precoder $\boldsymbol{F}$ that maximizes the minimum distance can be formulated as:

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} &= \arg\max_{\boldsymbol{F}} D^2_{\min}(\boldsymbol{HF}, \mathcal{A}) \\ &\text{subject to} \\ &\text{tr}(\boldsymbol{FF}^*) \leq P_0. \end{aligned} \tag{3.1}$$

Let $\boldsymbol{e} = \boldsymbol{a} - \tilde{\boldsymbol{a}} \in \mathcal{E}^B$, where $\mathcal{E}$ is the difference set of the alphabet $\mathcal{A}$. We now have $D^2_{\min}(\boldsymbol{HF}, \mathcal{A}) = \min_{\boldsymbol{e} \neq \boldsymbol{0}_B} \boldsymbol{e}^* \boldsymbol{F}^* \boldsymbol{H}^* \boldsymbol{HFe}$. Define

$$\boldsymbol{G} \triangleq \boldsymbol{F}^* \boldsymbol{H}^* \boldsymbol{HF} \tag{3.2}$$

to be the *Gram matrix* of $\boldsymbol{HF}$. Then (3.1) becomes

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} &= \arg\max_{\boldsymbol{F}} \min_{\boldsymbol{e} \neq \boldsymbol{0}_B, \boldsymbol{e} \in \mathcal{E}^B} \boldsymbol{e}^* \boldsymbol{Ge} \\ &\text{subject to} \\ &\text{tr}(\boldsymbol{FF}^*) \leq P_0. \end{aligned} \tag{3.3}$$

It is readily seen that the optimal $\boldsymbol{F}_{\text{opt}}$ is such that it minimizes $\text{tr}(\boldsymbol{FF}^*)$ subject to a fixed constraint on the minimum distance, e.g., $\min_{\boldsymbol{e} \neq \boldsymbol{0}_B, \boldsymbol{e} \in \mathcal{E}^B} \boldsymbol{e}^* \boldsymbol{Ge} \geq 1$. Thus, we can rewrite (3.3) as

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} &= \arg\min_{\boldsymbol{F}} \text{tr}(\boldsymbol{FF}^*) \\ &\text{subject to} \\ &\min_{\boldsymbol{e} \neq \boldsymbol{0}_B, \boldsymbol{e} \in \mathcal{E}^B} \boldsymbol{e}^* \boldsymbol{Ge} \geq 1 \\ &\boldsymbol{G} = \boldsymbol{F}^* \boldsymbol{H}^* \boldsymbol{HF}. \end{aligned} \tag{3.4}$$

As before, let $\boldsymbol{H} = \boldsymbol{USV}^*$ be the singular value decomposition of $\boldsymbol{H}$. For flat fading MIMO channels, $h_{i,j}$ are i.i.d. complex-valued Gaussian variables,

which implies that the rank of $\boldsymbol{H}$ is $N = \min(N_t, N_r)$ with probability 1. From the definition of $\boldsymbol{G}$ in (3.2), it follows that $\boldsymbol{U}$ has no impact on $D^2_{\min}(\boldsymbol{HF}, \mathcal{A})$, and can therefore be removed at the receiver. Furthermore, the matrix $\boldsymbol{V}$ can be absorbed into $\boldsymbol{F}$ without changing the value of the objective function in (3.4). Only the $N \times N$ diagonal submatrix in $\boldsymbol{S}$, that contains the singular values, is of interest, since the other elements are zero. Hence, an equivalent model to (2.1) arises:

$$\boldsymbol{y} = \boldsymbol{SFa} + \boldsymbol{n}, \tag{3.5}$$

where $\boldsymbol{S}$ is a $N \times N$ diagonal matrix with non-zero diagonal entries, $\boldsymbol{F}$ is an $N \times B$ matrix subject to $\mathrm{tr}(\boldsymbol{F}^*\boldsymbol{F}) \leq P_0$ and $\boldsymbol{a}$ a $B \times 1$ vector. Further, $\boldsymbol{y}$ is now $N \times 1$ and so is $\boldsymbol{n}$. In total, the system in (2.1), where $\boldsymbol{x} = \boldsymbol{Fa}$, can WLOG be reduced to (3.5). $\boldsymbol{G}$ is now $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}$, a $B \times B$ matrix. It is further assumed that $B \leq N$, i.e., the number of spatially multiplexed data streams is determined by the rank. Thus, we can now rewrite (3.4) as

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} &= \arg\min_{\boldsymbol{F}} \mathrm{tr}(\boldsymbol{FF}^*) \\ &\text{subject to} \\ &\min_{\boldsymbol{e} \neq \boldsymbol{0}_B, \boldsymbol{e} \in \mathcal{E}^B} \boldsymbol{e}^*\boldsymbol{Ge} \geq 1 \\ &\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}. \end{aligned} \tag{3.6}$$

In [58], it was shown that solving (3.6) is an NP-hard problem. Thus, at first sight, this problem seems mathematically intractable, and finding a solution to it online amounts to solving an NP-hard problem. Therefore, the first natural approach is to construct suboptimal solutions to (3.6), as was done in [61, 65]. This is the rationale behind Section 3.2. Another possibility is to solve (3.6) by an algorithmic approach, which is the focus of Section 3.3. Careful investigations of the precoders obtained in Section 3.3 reveal interesting connections between the problem in (3.6) and lattice theory. This connection is studied in Chapter 4.

Since $\mathbf{G}$ is Hermitian, and thus a normal matrix, its eigendecomposition is

$$\mathbf{G} = \mathbf{QDQ}^*. \tag{3.7}$$

The factorization (3.7) is unique if the diagonal elements of $\mathbf{D}$ are ordered in a decreasing order. From the definition of $\boldsymbol{G}$ and (3.7), we see that $\mathbf{F} = \mathbf{S}^{-1}(\sqrt{\mathbf{D}}\ \mathbf{0}_{B,N-B})^{\mathrm{T}}\mathbf{Q}^*$ is a precoder such that $\mathbf{F}^*\mathbf{S}^2\mathbf{F} = \mathbf{G}$. The $\mathbf{0}_{B,N-B}$ matrix is an $B \times N - B$ zero matrix, accounting for the case when $B < N$. Next we prove that this $\mathbf{F}$ has the lowest energy of all possible $\mathbf{G}$ satisfying $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}$.

Theorem 7. *Let* $\mathbf{G} = \mathbf{QDQ}^*$ *where the diagonal elements of* $\mathbf{D}$ *are ordered in decreasing order. Then, of all* $\mathbf{F}$ *satisfying* $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}$*,*

$$\mathbf{F} = \mathbf{S}^{-1}(\sqrt{\mathbf{D}}\ \mathbf{0}_{B,N-B})^{\mathrm{T}}\mathbf{Q}^* \tag{3.8}$$

is the one with least energy $\mathrm{tr}(\mathbf{F}^*\mathbf{F})$*.*

Proof: Combining $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}$ and (3.7) we get $\mathbf{F}^*\mathbf{S}^2\mathbf{F} = \mathbf{QDQ}^*$. Rewriting, we have

$$\boldsymbol{Q}^*\mathbf{F}^*\mathbf{S}_{\boldsymbol{H}}^2\mathbf{F}\boldsymbol{Q} = \mathbf{D}. \tag{3.9}$$

Assume that $\mathbf{F}^*\mathbf{R}\mathbf{F}$, where $\mathbf{R}$ is a positive semidefinite matrix, is equal to a diagonal matrix $\mathbf{\Sigma}$, where the diagonal elements in $\mathbf{\Sigma}$ are in decreasing order (in our case $\mathbf{\Sigma}$ is $B \times B$). Then, in [49, Lemma 3.16] it is proved that we can always choose $\mathbf{F} = \mathbf{V}_{\mathbf{R}}\mathbf{D}_{\mathbf{R}}^{-1/2}\sqrt{\mathbf{\Sigma}}$, where $\mathbf{V}_{\mathbf{R}}$ contains the B eigenvectors of $\mathbf{R}$ corresponding to the B largest eigenvalues of $\mathbf{R}$ and $\mathbf{D}_{\mathbf{R}}$ contains the B largest eigenvalues of $\mathbf{R}$, respectively, in order to minimize $\mathrm{tr}(\mathbf{F}^*\mathbf{F})$. Hence in our case, we choose $\mathbf{F}\boldsymbol{Q} = \mathbf{S}^{-1}(\mathbf{I}_{\mathbf{B}\times\mathbf{B}}\,\mathbf{0}_{B,N-B})^{\mathrm{T}}\sqrt{\mathbf{D}}$ which gives $\mathbf{F} = \mathbf{S}^{-1}(\sqrt{\mathbf{D}}\,\mathbf{0}_{B,N-B})^{\mathrm{T}}\boldsymbol{Q}^*$ and completes the proof.

3.2 Suboptimal Constructions

We start by investigating efficient suboptimal solutions to (3.6). Since the precoders are suboptimal, they should provide another advantage, such as low decoding complexity. This is the idea behind our suboptimal construction presented in this section. Throughout the section, we will assume that $\mathcal{A}$ is a QPSK alphabet. Thus, since $\mathcal{A}$ is fixed, we will write $D^2_{\min}(\boldsymbol{HF})$ instead of $D^2_{\min}(\boldsymbol{HF}, \mathcal{A})$.

3.2.1 Relaxation into a Toeplitz G

We constrain the Gram matrix $\mathbf{G}$ to a Hermitian Toeplitz form

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & \cdots & g_{B-1} \\ g_1^* & g_0 & g_1 & \ddots & & \vdots \\ g_2^* & g_1^* & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & g_1 & g_2 \\ \vdots & & \ddots & g_1^* & g_0 & g_1 \\ g_{B-1}^* & \cdots & \cdots & g_2^* & g_1^* & g_0 \end{pmatrix}. \tag{3.10}$$

Constraining $\mathbf{G}$ to a Toeplitz matrix is clearly a suboptimal solution to the problem in (3.6); however, it admits analytical treatment while, as we will demonstrate, the results are still satisfying. The reasons that motivate us to pursue a Toeplitz structure are as follows. Assume that $\mathbf{G}$ has diagonal elements $g_{1,1} \neq g_{2,2} \neq ... \neq g_{B,B}$, and consider the resulting Euclidean distance from error vectors with only a single non-zero entry. Within the class of such error vectors, the minimum distance equals[1] $4\min(g_{1,1}, ..., g_{B,B})$ achieved by an error vector with its non-zero entry at the position for the minimum of the diagonal elements. Thus, the minimum distance is completely determined from the smallest diagonal element. Thus, we gain nothing from having many large diagonal elements if there exists a single small one. On the contrary, large diagonal elements should be avoided if possible since they are in general expensive in terms of the cost constraint $\mathrm{tr}(\mathbf{F}^*\mathbf{F})$; the design should aim at having a well balanced $\mathbf{G}$. Now assume error vectors with exactly q non-zero entries and consider a sub block $\boldsymbol{G}_q$ of size $q \times q$ formed from the rows and columns $p, \ldots, p+q-1$ of $\boldsymbol{G}$, i.e., $\boldsymbol{G}_q = g_{p:p+q-1,p:p+q-1}$, for some p. Invoking the same arguments as above leads to the conclusion that all such blocks (for different p) should be identical, the worst block will determine the MSED. We would like to strongly point out that these are not hard facts, but merely general design rules that lead to a tractable problem.

Using the precoder $\mathbf{F}$ from Theorem 7 gives that

$$\mathrm{tr}(\mathbf{F}^*\mathbf{F}) = \mathrm{tr}(\mathbf{S}^{-2}\mathbf{D}) = \sum_{j=1}^{B} \frac{d_{j,j}}{s_{j,j}^2}, \tag{3.11}$$

where $d_{j,j}$ and $s_{j,j}$ denote the j:th diagonal element in $\mathbf{D}$ and $\mathbf{S}$, respectively. Now we can rewrite the original problem in (3.6) in a simpler way. Using (3.11), we can reformulate (3.6) as

$$\min_{\boldsymbol{d}} \sum_{j=1}^{B} \frac{d_{j,j}}{s_{j,j}^2}, \quad \text{subject to} \begin{cases} \min_{\boldsymbol{e}\neq\mathbf{0}} \boldsymbol{e}^*\mathbf{G}\boldsymbol{e} \geq 1 \\ d_{j,j} > 0, \ j = 1, \ldots, B. \end{cases} \tag{3.12}$$

Hence, the objective function in (3.12) is linear in the eigenvalues of $\mathbf{G}$; unfortunately, the eigenvalues are in general hard to express in the elements $g_{i,j}$ of $\mathbf{G}$.

We will next study two different Toeplitz structures where the eigenvalues can be found in closed form and be expressed in the elements of $\boldsymbol{G}$; a memory-1 structure and a cyclic structure of memory-4. The latter gives rise to a cyclic

[1]The elements with least energy in the error alphabet are ± 2 and $\pm 2i$ which all have energy 4.

Toeplitz $\boldsymbol{G}$, while the former is not cyclic. Imposing a memory-1 structure or a cyclic Toeplitz structure on $\boldsymbol{G}$ makes (3.12) a simple optimization problem for any $\boldsymbol{S}$. The memory of $\boldsymbol{G}$ is related to the number of non-zero entries in $\boldsymbol{G}$, which determines the ML decoding complexity at the receiver: higher memory gives larger decoding complexity. Still, increasing the memory gives more degrees of freedom for the optimization in (3.12), and thus better precoders are expected to be found.

3.2.2 Memory-1 Toeplitz structure

In this section we set $g_k = 0$, $k > 1$. An ML detector can be be implemented by using a Cholesky-factorization $\mathbf{L}^*\mathbf{L} = \mathbf{G}$; see [46] for the details. Due to the constraint $g_k = 0$, $k > 1$ it follows that $L_{k-\ell} = 0$ unless $k - \ell = 0$ or 1. This implies that the memory of the associated ML-detector is unity and ML-detection can be implemented by a memory-1 Viterbi algorithm.

The eigenvalues $d_{k,k}$ of such a matrix can be shown to equal

$$d_{k,k} = g_0 + 2|g_1| \cos\left(\frac{k\pi}{B+1}\right), \quad k = 1, \ldots, B. \tag{3.13}$$

Clearly, $\{d_{k,k}\}$ is a decreasing sequence which implies that factorization (3.7) becomes unique. Inserting this into the expression for the objective function in (3.12), we get

$$\sum_{k=1}^{B} \frac{g_0 + 2|g_1| \cos\left(\frac{k\pi}{B+1}\right)}{s_{k,k}^2}.$$

Hence, the objective function in (3.12) becomes linear in g_0 and $|g_1|$. It can without loss of generality be assumed that $g_0 = 1$. Since $d_{k,k} > 0$ in (3.13), it holds that $0 < |g_1| < -1/2 \cos(B\pi/(B+1))$ (which tends to $1/2$ as B grows).

There exists an interesting connection between the problem studied here and minimum distance problem for intersymbol interference (ISI) channels. If $|g_1| \leq 1/2$ then $\mathbf{G}$ can be interpreted as the auto-correlation matrix of a memory-1 ISI channel. It can be proved [64] that for all memory-1 ISI channels the MSED is achieved by an error event consisting of a single symbol and equals $4g_0$. However, we do not require $\mathbf{G}$ to be a valid auto-correlation matrix, but only to be positive definite. Therefore, a $|g_1|$ that exceeds $1/2$ can be used.

The problem in (3.12) was solved exhaustively for 100 000 different channel realizations, and an interesting observation can be made: The optimal solution was always that $|g_1|$ should be chosen as large as possible so that $D_{\min}^2(\boldsymbol{SF}, \mathcal{A})$ is caused by a single symbol error event and thus equals $4g_0$. That large $|g_1|$ are beneficial for the objective function can be seen directly from (3.12); since the

sequence $\{1/s_{k,k}^2\}$ is ordered in increasing order, $\cos\left(\frac{k\pi}{B+1}\right) = -\cos\left(\frac{(B-k)\pi}{B+1}\right)$ for $k = 1 \ldots \lfloor B/2 \rfloor$ and $\{d_{k,k}\}$ is a decreasing sequence, it follows that the objective function in (3.12) is minimized by choosing $|g_1|$ as large as possible. However, when $|g_1|$ exceeds a certain threshold, the MSED drops below the single error symbol distance $4g_0$, and the decrease of the MSED turns out to be more significant than the decrease of the objective function. Table 3.1 lists the g_1 that have as large magnitude as possible such that the MSED is generated from an error event consisting of a single error symbol.

Table 3.1: Optimal g_1 for some values of B.

B	g_1
6	$0.5545\, e^{i\pi 3/20}$
8	$0.5320\, e^{i\pi 3/20}$
10	$0.5210\, e^{i\pi 1/5}$

3.2.3 Memory-4 cyclic Toeplitz structure

In this case we use a $\mathbf{G}$ of the form (3.10) with $g_{B-1} = g_1^*$, $g_{B-2} = g_2^*$ and $g_k = 0$, $3 \leq k \leq B-3$. Note that a cyclic memory-2 construction has the same degrees of freedom as the memory-1 construction in Section 3.2.2, since only g_1 can be chosen at will, while a cyclic memory-3 construction is not possible since $\boldsymbol{G}$ should be Cyclic Toeplitz (thus, memories that are odd numbers are not possible). The reason for constraining ourselves to a cyclic structure is because it is possible to obtain an analytic expression for the eigenvalues $\{d_{k,k}\}$, while still maintaining a low decoding complexity. Note that compared with the $\mathbf{G}$ in Section 3.2.2, the memory-4 cyclic $\mathbf{G}$ has extra elements in the upper right and lower left corners. This increases the complexity of ML-detection to 4, which can be realized by viewing the system as a tailbiting system of memory 2.

Since $\mathbf{G}$ is a $B \times B$ cyclic Toeplitz matrix, we know from [31] that the $\boldsymbol{Q}$ in (3.7) is the DFT matrix, that is, the element at position (k, ℓ) in $\boldsymbol{Q}$ equals $e^{-2\pi i(k-1)(\ell-1)/B}/\sqrt{B}$. Moreover, the diagonal elements of $\mathbf{D}$ are the Inverse DFT (IDFT) of the first row of $\mathbf{G}$, i.e., $\boldsymbol{d} = \mathrm{IDFT}(g_0, g_1, \ldots, g_{B-1})$, where $\boldsymbol{d} = (d_{1,1}, d_{2,2}, \ldots, d_{B,B})$ is the main diagonal of $\mathbf{D}$. The idea of freezing the unitary matrix into a DFT matrix and only optimizing the diagonal matrix $\boldsymbol{D}$ has also been exploited in [66], but the linear optimization to follow shortly was not used in [66]. Instead, a suboptimal static power allocation was used. Important to note here is that every cyclic Toeplitz matrix can be obtained by

simply choosing corresponding diagonal elements in $\mathbf{D}$; $\boldsymbol{Q}$ is the same for all cyclic Toeplitz matrices. Consequently the optimization problem (3.12) turns into an optimization problem over $\boldsymbol{d}$. Also, from (3.7) it follows that

$$\boldsymbol{e}^*\mathbf{G}\boldsymbol{e} = \tilde{\boldsymbol{e}}^*\mathbf{D}\tilde{\boldsymbol{e}} = \sum_{j=1}^{B} |\tilde{e}_j|^2 d_{j,j}, \tag{3.14}$$

where $\tilde{\boldsymbol{e}} = \boldsymbol{Q}^*\boldsymbol{e}$. Hence $\tilde{\boldsymbol{e}}$ is the DFT of the error vector $\boldsymbol{e}$. Since $g_k = 0$, $3 \leq k \leq B-2$, we have an additional constraint on $\boldsymbol{d}$. Now we can rewrite (3.12) as

$$\min_{\boldsymbol{d}} \sum_{j=1}^{B} \frac{d_{j,j}}{s_{j,j}^2}, \quad \text{subject to} \begin{cases} \min_{\tilde{\boldsymbol{e}}\neq\mathbf{0}} \sum_{j=1}^{B} |\tilde{e}_j|^2 d_{j,j} \geq 1 \\ d_{j,j} > 0, \ j = 1, \ldots, N. \\ \sum_{j=1}^{B} d_{j,j} e^{2\pi i (j-1)k/B} = 0, \\ \quad 3 \leq k \leq B-3. \end{cases} \tag{3.15}$$

Note that the problem is linear in the eigenvalues $\boldsymbol{d}$, and is thus efficiently solvable by, e.g., the simplex method. However, the number of constraints in (3.15) is of an exponential order; in our case roughly 9^B. Hence, it is of interest to somehow reduce the number of constraints in (3.15), while still finding a precoder that is optimal or very close to optimal. Therefore, a simple algorithm is devised that reduces the number of constraints. It is summarized in the following flowchart:

1. Start with an initial set $\mathcal{E}$ of vectors $\boldsymbol{e}$.
2. Solve problem (3.12) over $\mathcal{E}$. Let $\boldsymbol{d}_{\text{opt}}$ denote the solution.
3. Use $\boldsymbol{d}_{\text{opt}}$ to construct $\mathbf{G}$ by means of (3.8) and compute the MSED. Let $\boldsymbol{e}_{\text{opt}}$ denote the worst error event.
4. If $\boldsymbol{e}_{\text{opt}} \in \mathcal{E}$, stop: $\boldsymbol{d}_{\text{opt}}$ produces maximal MSED. Otherwise, add $\boldsymbol{e}_{\text{opt}}$ to $\mathcal{E}$, return to step 2.

The above algorithm was run for 100000 channel realizations for each antenna combination. The initial list was initialized with all error vectors of form $\boldsymbol{e} = [0 \ \ldots \ 0 \ e_0 \ e_1 \ e_2 \ 0 \ \ldots \ 0]^{\mathrm{T}}$. From the experiment, the following two things are observed. (i) The final list $\mathcal{E}$ (after 100000 channel realizations) is not much larger than the initial list; for most antenna combinations 5-10 new error vectors were added. (ii) The above algorithm produces very few different results $\boldsymbol{d}$; two vectors $\boldsymbol{d}$ and $\tilde{\boldsymbol{d}}$ are declared identical if $\|\boldsymbol{d} - \tilde{\boldsymbol{d}}\|^2 \leq 10^{-4}$. For example, only 2 different $\boldsymbol{d}$:s were found when $N = 8$, $B = 6$ and 4 different when $N = 12$, $B = 10$; in the latter case, 2 of the 4 were used in 99.9% of the cases.

Interestingly, it turns out that the optimal $\boldsymbol{d}$ is usually not a decreasing sequence. Thus, Theorem 7 implies that the obtained $\boldsymbol{d}$ is not optimal to use when constructing the precoder. The reason for the non-decreasing $\boldsymbol{d}$ is that $\boldsymbol{d}$ is optimal for the (DFT) $\boldsymbol{Q}$ under investigation, but $\boldsymbol{Q}$ is not the best unitary matrix to use. In order to improve, we construct $\mathbf{G}$ from the DFT $\boldsymbol{Q}$ and its optimal $\mathbf{D}$ (which has the optimal $\boldsymbol{d}$ on its diagonal). Then a unique eigendecomposition follows (which results in a non-DFT $\boldsymbol{Q}$) and the optimal precoder is constructed from this composition, according to Theorem 7. The newly obtained $\mathbf{D}$ matrix is simply a reordering of the optimal $\boldsymbol{d}$ that comes from the optimization, so that its diagonal elements are sorted in decreasing order.

Since the optimization in (3.12) is considered, all $\boldsymbol{d}$ are constrained to produce a fixed MSED that equals 1. Thus, to pick the best precoder for a given channel realization, one has merely to identify the vector $\boldsymbol{d}$ from the list that results in the least energy consumption $\sum d_{j,j}/s_{j,j}^2$. Since the list is short this involves almost no complexity. Hence, these precoders are as easy to construct as closed form precoders.

3.2.4 Comparisons

We will now compare the results of our precoder design with the competing design in [65], which is described next. Let

$$\mathbf{S} = \begin{pmatrix} s_{1,1} & 0 \\ 0 & s_{2,2} \end{pmatrix}$$

be a 2×2 channel matrix. In [67], the *optimal* precoder for a 2×2 MIMO channel is found. Let $s_{1,1} = \rho \cos\mu$, $s_{2,2} = \rho \sin\mu$, where $0 \leq \mu \leq \pi/4$. Then the optimal precoder $\boldsymbol{F}_{\text{opt}}$ is given by

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} &= \begin{pmatrix} \sqrt{\frac{3+\sqrt{3}}{6}} & \sqrt{\frac{3-\sqrt{3}}{6}}\, e^{i\pi/12} \\ 0 & 0 \end{pmatrix}, \ 0 \leq \mu \leq \mu_0 \\ \boldsymbol{F}_{\text{opt}} &= \sqrt{\frac{1}{2}} \begin{pmatrix} \cos\phi & 0 \\ 0 & \sin\phi \end{pmatrix} \begin{pmatrix} 1 & e^{i\pi/4} \\ -1 & e^{i\pi/4} \end{pmatrix}, \ \mu_0 \leq \mu \leq \frac{\pi}{4} \end{aligned} \quad (3.16)$$

where $\mu_0 \approx 17.28^\circ$ and $\phi = \arctan\left(\frac{\sqrt{2}-1}{\cos\mu}\right)$. In [65], the 2×2 results in (3.16) are used as building blocks to construct large precoders for a general $N \times N$ MIMO setup as in (3.5) (thus, $\boldsymbol{S}$ is now $N \times N$, where N is assumed to be an even integer). The construction is as follows:

1. Pair the largest and smallest singular values of the channel $\mathbf{S}$, i.e., create the following $N/2$ pairs:

$$(s_{1,1}, s_{N,N}), (s_{2,2}, s_{N-1,N-1}), \ldots, (s_{N/2,N/2}, s_{N/2+1,N/2+1}).$$

2. Consider each pair as a 2×2 MIMO channel and construct the optimal precoder for it according to (3.16). Let $\boldsymbol{F}_i$ be the optimal precoder for pair i.

3. The available energy is distributed across the $\{\boldsymbol{F}_i\}$ so that $\mathrm{tr}(\boldsymbol{F}_i^* \boldsymbol{F}_i) = [D^2_{\min,i}(\boldsymbol{S}\boldsymbol{F}_i) \sum_k 1/D^2_{\min,k}(\boldsymbol{S}\boldsymbol{F}_k)]^{-1}$, where $D^2_{\min,i}(\boldsymbol{S}\boldsymbol{F}_i)$ is the squared minimum distance of precoder $\boldsymbol{F}_i$. This energy distribution makes the minimum distance of all the 2×2 channel pairs equal.

Denote the joint precoder matrix formed from all $\{\boldsymbol{F}_i\}$ as $\boldsymbol{P}_{\mathrm{f}}$.

In Figure 3.1 we plot the pdf of $D^2_{\min}(\boldsymbol{S}\boldsymbol{F})$ for different precoders $\boldsymbol{F}$: the memory-1 precoder from Section 3.2.2, the memory-4 precoder from Section 3.2.3 and the precoder $\boldsymbol{P}_{\mathrm{f}}$ from [65]. The figure is for the case when $B = N-2$. We clearly see that the memory-4 precoder is superior, followed by memory-1 and $\boldsymbol{P}_{\mathrm{f}}$. The gain is significant and will also be present in receiver tests to follow. Figure 3.2 shows the distance profile when $B = N - 1$. Still, the outcome is the same as when $B = N - 2$: memory 4 has the best distance profile, followed by memory 1 and $\boldsymbol{P}_{\mathrm{f}}$. However, the difference is not as big as when $B = N - 2$. This can be somewhat explained by the fact that including the second smallest singular value of the channel (when $B = N - 1$ we use the $N - 1$ largest singular values of the N possible) adds one more term in the objective function in (3.12), namely $d_{B-1,B-1}/s^2_{B-1,B-1}$, where $s_{B-1,B-1} \leq s_{B-2,B-2} \leq \ldots \leq s_{1,1}$. This will increase the value of (3.12), which is the energy $\mathrm{tr}(\boldsymbol{F}^*\boldsymbol{F})$ of the precoder, and hence the $D^2_{\min}(\boldsymbol{S}\boldsymbol{F}) = 1/\mathrm{tr}(\boldsymbol{F}^*\boldsymbol{F})$ will become smaller than when $B = N - 2$.

Next, we compare the symbol error rate (SER) performance of our proposed precoders in Sections 3.2.2 and 3.2.3 with the precoder $\boldsymbol{P}_{\mathrm{f}}$ from [65]. Figure 3.3 shows the comparison between the different precoders. SNR in the figure is defined as $\mathrm{SNR} = 1/N_0$. We can clearly see from the figure that the memory-4 precoder is the best one in terms of SER, followed by the memory-1 and $\boldsymbol{P}_{\mathrm{f}}$. This is in agreement with Figure 3.1, which dictates the SER behaviour for large SNR values and predicts the results observed in Figure 3.3. Note that the performance of our precoders is better when we increase B and the dimension N of the MIMO system.

Figure 3.4 shows the comparison between the precoders when $B = N - 1$. As expected from Figure 3.2, the SER curve difference is only marginal in this case. When $B = N$, $\boldsymbol{P}_{\mathrm{f}}$ actually outperformed our precoders. We observed

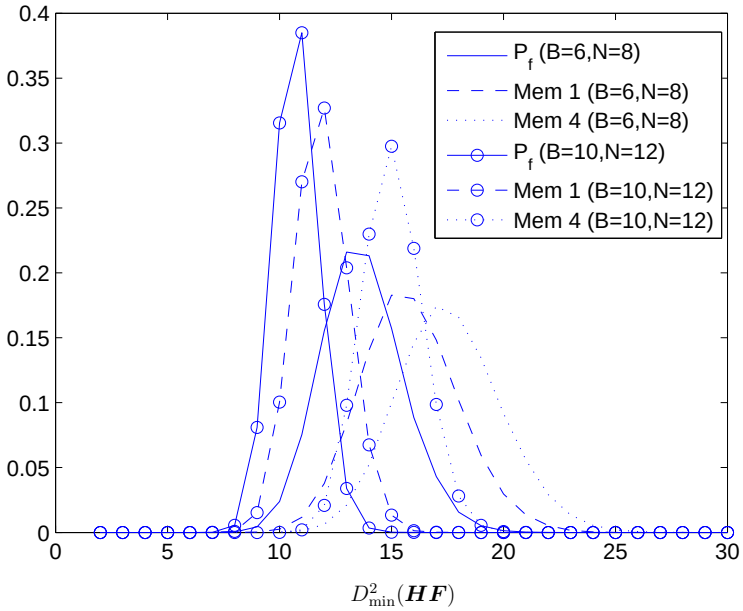

Figure 3.1: Pdf of $D^2_{\min}(\boldsymbol{HF})$ for various precoders. $B = N - 2$.

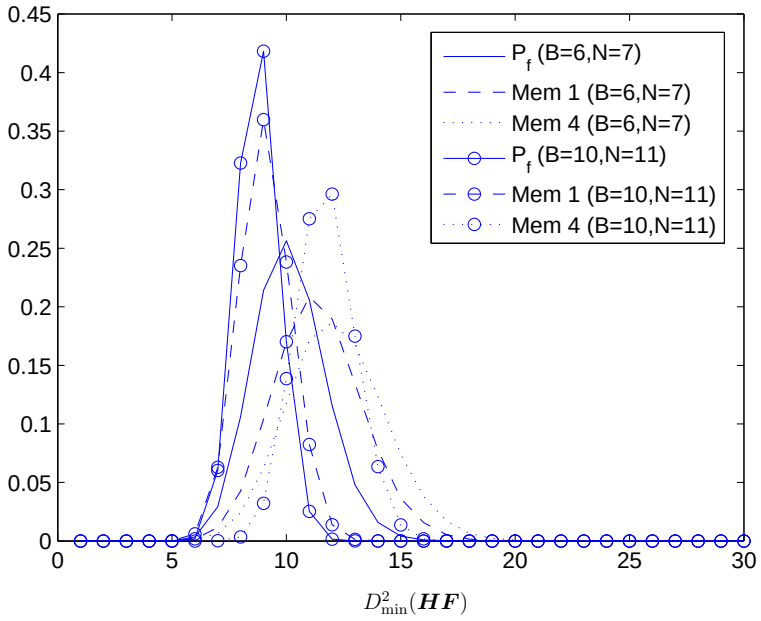

Figure 3.2: Pdf of $D^2_{\min}(\boldsymbol{HF})$ for various precoders $\boldsymbol{F}$. The size of $\boldsymbol{G}$, which equals the number of data streams, is $B = N - 1$.

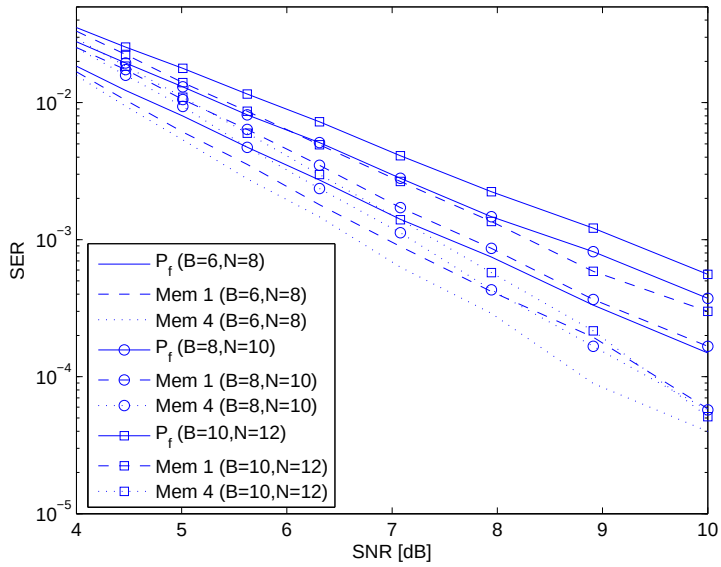

Figure 3.3: SER comparisons when $B = N - 2$. Solid lines show the performance of $\boldsymbol{P}_{\mathrm{f}}$, which is the competing scheme from [65]. The dashed lines stand for the performance of the precoders that give a memory-1 structure on $\boldsymbol{G}$, while the dotted ones show the performance of the precoders that give a memory-4 cyclic structure on $\boldsymbol{G}$.

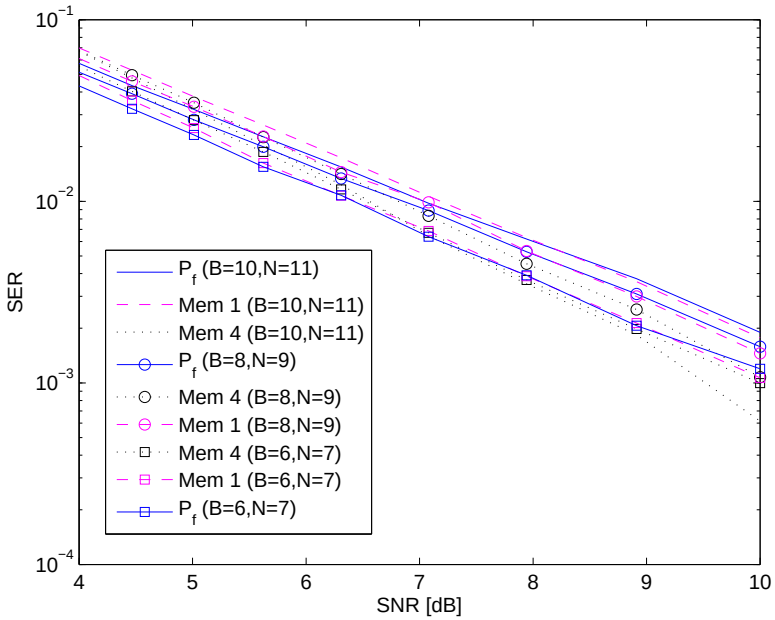

Figure 3.4: SER comparisons when $B = N - 1$. The solid line shows the performance of the competing scheme from [65]. The dashed lines stand for the performance of the precoders that produce a memory-1 $\boldsymbol{G}$, while the dotted ones show the performance of the precoders that produce a memory-4 cyclic $\boldsymbol{G}$.

that indeed, the function in (3.12) became large, hence the minimum distance advantage for $B = N - 1$ and $B = N - 2$ is gone.

3.3 Iterative Precoder Optimization

The preceding section presented a suboptimal construction of $\boldsymbol{F}$, which is efficient in terms of complexity and SER performance. It was derived by imposing a specific structure on the Gram matrix $\boldsymbol{G}$, from which it is also easy to control the ML decoding complexity at the receiver. The aim is now to drop this constraint on $\boldsymbol{G}$, and perform an optimization that produces even better precoders. Herein, we present an iterative optimization method, which alternates between optimizing the unitary matrix $\boldsymbol{Q}$ and the eigenvalues $\boldsymbol{D}$ of $\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^*$. We focus on the case $B = N$, with the remark that the optimization is easily applied to $B \neq N$ as well. Hence, the Gram matrix $\boldsymbol{G}$ is now $N \times N$.

Our optimization procedure alternates between optimization of $\boldsymbol{D}$ and $\boldsymbol{Q}$. The steps are the following:

1. Given a unitary matrix $\boldsymbol{Q}$, optimize (3.6) over $\boldsymbol{D}$.
2. With the obtained $\boldsymbol{D}$, optimize (3.6) over $\boldsymbol{Q}$.
3. Iterate the first two steps, until the increase in the value of the objective function $D^2_{\min}(\boldsymbol{H}\boldsymbol{F})$ becomes negligible.

Next, each of these optimization steps will be explained in detail.

3.3.1 Optimization over $\boldsymbol{D}$

For a fixed $\boldsymbol{Q}$, the minimum distance constraint in (3.6) can be written as

$$\min_{\boldsymbol{e}} \boldsymbol{e}^* \boldsymbol{G} \boldsymbol{e} = \min_{\tilde{\boldsymbol{e}}} \tilde{\boldsymbol{e}}^* \boldsymbol{D} \tilde{\boldsymbol{e}},$$

where $\tilde{\boldsymbol{e}} \stackrel{\triangle}{=} \boldsymbol{Q}^* \boldsymbol{e}$, and thus the optimization (3.6) reduces to

$$\begin{gathered} \min_{\{d_{j,j}\}, d_{j,j}>0, j=1,\ldots,N} \sum_{m=1}^{N} s_{m,m}^{-2} d_{m,m} \\ \text{subject to} \\ \tilde{\boldsymbol{e}}^* \boldsymbol{D} \tilde{\boldsymbol{e}} \geq 1, \quad \forall\, \tilde{\boldsymbol{e}}. \end{gathered} \tag{3.17}$$

Hence, the optimization is a linear problem over $\boldsymbol{D}$ and can be trivially solved by means of standard techniques, such as the simplex algorithm [68].

3.3.2 Optimizing for the unitary matrix Q

For a fixed $\boldsymbol{D}$, the constraint in (3.3) is fulfilled and thus only the objective function in (3.3) has to be optimized over $\boldsymbol{Q}$. The problem of optimizing objective functions under the unitary matrix constraint has been extensively treated in [69, 70]. In our case, the unitary optimization problem is:

$$\mathbf{Q}_{\text{opt}} = \arg\max_{\mathbf{Q}} \min_{\mathbf{e}} \mathbf{e}^* \mathbf{Q}\mathbf{D}\mathbf{Q}^* \mathbf{e}, \quad \text{subject to } \boldsymbol{Q}^*\boldsymbol{Q} = \mathbf{I}_{N\times N}. \tag{3.18}$$

For a given N and symbol constellation, there is a finite number of error vectors $\mathbf{e} \in \{\mathbf{e}_1, \ldots, \mathbf{e}_L\}$. For an M^2-QAM alphabet $\mathcal{A}$, the size of $\mathcal{E}^N$ is $|\mathcal{E}^N| = L = (2M-1)^N - 1$. This set can be reduced by removing $\boldsymbol{e}_j$ that can be expressed as $\pm\boldsymbol{e}_k$ or $\pm i\boldsymbol{e}_{\boldsymbol{k}}$ for some $k \neq j$.

The optimization problem (3.18) is equivalent to maximizing the minimum of a set of L continuously-differentiable objective functions $\mathcal{J}_\ell(\boldsymbol{Q})$ over the Lie group of unitary matrices $U(N)$:

$$\boldsymbol{Q}_{\text{opt}} = \arg\max_{\boldsymbol{Q}} \min_{\ell} \{\mathcal{J}_\ell(\boldsymbol{Q})\}_{\ell=1}^{L}, \quad \mathbf{Q} \in U(N), \tag{3.19}$$

where each of the objective functions is defined as

$$\mathcal{J}_\ell(\boldsymbol{Q}) = \mathbf{e}_\ell^* \boldsymbol{Q}\mathbf{D}\boldsymbol{Q}^* \mathbf{e}_\ell, \quad \ell = 1, \ldots, L. \tag{3.20}$$

Since the number of error vectors L is considerably large (order of thousands), the complexity of the optimization problem needs to be reduced. Without loss of generality, we consider the set of vectors $\{\mathbf{e}_\ell\}$ being sorted in an ascending order, according to the values of the objective function $\mathcal{J}_m$ they produce, i.e., $\mathcal{J}_1 \leq \mathcal{J}_2 \leq \ldots \leq \mathcal{J}_L$. Maximizing the minimum value of the inner objective function $\mathcal{J}_\ell$, is equivalent to maximizing $\mathcal{J}_1 = \min_\ell \{\mathcal{J}_\ell(\boldsymbol{Q})\}_{\ell=1}^{L}$ w.r.t $\boldsymbol{Q} \in U(N)$. The optimization w.r.t. the unitary matrix $\boldsymbol{Q}$ is done by using the Riemannian Steepest Ascent (SA) algorithm on the unitary group given in [70, Table I]. After each iteration of the SA algorithm, the vectors $\{\mathbf{e}_\ell\}$ are again sorted in ascending order, and the new obtained objective function $\mathcal{J}_1(\boldsymbol{Q})$ is maximized. The Euclidean gradient of $\mathcal{J}_1(\boldsymbol{Q})$ at a point $\boldsymbol{Q}_k \in U(N)$ is given by $\boldsymbol{\Gamma}_{1|k} = \mathbf{e}_1\mathbf{e}_1^*\boldsymbol{Q}_k\mathbf{D}$, and represents the steepest ascent direction on the Euclidean space. The Riemannian gradient is a skew-Hermitian matrix $\mathbf{Y}_{1|k} = \boldsymbol{\Gamma}_{1|k}\boldsymbol{Q}_k^* - \boldsymbol{Q}_k\boldsymbol{\Gamma}_{1|k}^*$, and represent the steepest ascent direction on the constrained parameter space $U(N)$ at $\boldsymbol{Q}_k$, translated to a group identity element. A rotational update is performed, such that the unitary matrix constraint is maintained at every iteration:

$$\boldsymbol{Q}_{k+1} = \exp(+\alpha\mu\mathbf{Y}_{1|k})\boldsymbol{Q}_k, \tag{3.21}$$

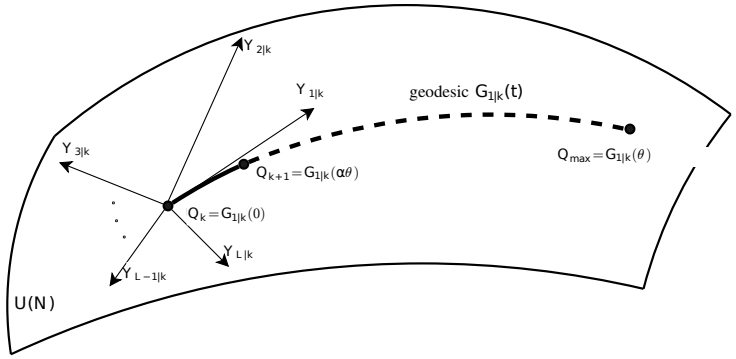

Figure 3.5: Optimization of $\mathcal{J}_1(\boldsymbol{Q})$ on the unitary Lie group $U(N)$. The Lie group $U(N)$ can be viewed as a smooth curved surface determined by the unitary constraint. In this space, geodesics are the equivalent of the straight lines from the Euclidean space. At each iteration k, the algorithm moves along the geodesic curve $\boldsymbol{\mathcal{G}}_{1|k}(t)$ emanating from the current point $\boldsymbol{Q}_k \in U(N)$ in the direction of Riemannian gradient $\mathbf{Y}_{1|k}$. Only a fraction $\alpha \in (0,1)$ of the complete step μ (that would produce the maximum increase of $\mathcal{J}_1(\boldsymbol{Q})$) is taken, at each iteration.

where $\exp(\cdot)$ is the standard matrix exponential[2], and μ is the step size. Note that the update is multiplicative since a product of unitary matrices is unitary.

The complexity of the unitary optimization itself is of order $\mathcal{O}(N^3)$. However, the complextiy of sorting the error vectors $\{\mathbf{e}_\ell\}$ in ascending order is $\mathcal{O}(LN^2)$. Since $N \ll L$, the complexity of the entire optimization is therefore dominated by the sorting operation. Techniques to reduce the number of error vectors $\{\mathbf{e}_\ell\}$ exist, such as the flowchart in Section 3.2.3 and the method in [71]. These have not been used since with a standard work-station, the entire optimization process is a matter of fractions of a second.

A highly accurate step size μ is selected by using the polynomial-based line search method given in [69, Table 1]. The unitary optimization procedure on $U(N)$ is illustrated in Figure 3.5.

The scaling factor $\alpha \in (0,1)$ prevents the objective function $\mathcal{J}_1(\boldsymbol{Q})$ to increase too quickly. Too much increase in $\mathcal{J}_1(\boldsymbol{Q})$ may actually produce a decrease in the other objective functions $\{\mathcal{J}_\ell(\boldsymbol{Q})\}_{\ell=2}^{L}$, even below the initial value of $\mathcal{J}_1$. This is because their corresponding gradients $\mathbf{Y}_{m|k}$ do not necessarily point in directions that increase the minimum value of the objective functions.

[2]The matrix $\exp(+\alpha\mu\mathbf{Y}_{1|k})$ is a unitary matrix.

In that case, instead of increasing the minimum value of $\{\mathcal{J}_\ell(\boldsymbol{Q})\}_{\ell=1}^{L}$, the value would be decreased. Therefore, small steps are preferred in order to avoid this behavior[3]. After every step, reordering the error vectors $\{\mathbf{e}_\ell\}$ in required, in order to maximize the minimum value of the objective functions, $\mathcal{J}_1$.

3.4 Optimization Results

The convergence properties of the iterative optimization method described in Section 3.3 depends heavily on the step size μ. The optimization shows rapid convergence whenever μ is moderate-large, however, it often converges to a local optimum. To improve the results, the step size μ should be taken as a small number. This unfortunately implies that several hundreds of iterations are needed before saturation of the objective function is reached.

It turns out that the starting point does not have a significant effect whenever μ is small. For each $\boldsymbol{S}$ in the grid, we have chosen as starting point the optimal precoder for the previously considered $\boldsymbol{S}$, which is close in Euclidean distance to the current $\boldsymbol{S}$, but also, 10 randomly chosen starting points. Then we take as output the best of the 11 solutions, but most often they are all the same. The overall conclusion of the iterative optimization is that it is efficient, but with a remark that it should be carried out off-line since several hundreds of iterations are needed. If suboptimal solutions are tolerated, μ can be taken larger which results in faster convergence so that the optimization can be carried out on-line.

It may appear problematic to carry out the optimization off-line since the size of precoder codebook may be prohibitive. However, for $\boldsymbol{S}$ that are "close", the optimal Gram matrices $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}$ are scaled versions of each other. The size of the codebook to choose from becomes in fact quite small. The same is not true for the optimal mutual information precoders from [83]. In that case, no precoder codebook can be tabulated since each channel $\boldsymbol{S}$ has a unique optimal precoder. This is a strong motivation to consider minimum distance optimal precoders.

As a final remark, a standard Gauss-Seidel optimization[4] of the precoder $\boldsymbol{F}$ was tested, but it turns out that such an approach is grossly inferior to the iterative optimization. This is true both for running-time and accuracy of the results, and most often an undesired local optimum is reached.

Next, the outcome of the optimization procedure for the considered setups is described. The optimal $\boldsymbol{G}$ matrices are not listed in the thesis, but they can

[3]The reason why steepest ascent is used is that the conjugate gradient in [70] would be "too fast" for this purpose.

[4]From the optimization toolbox in MATLAB.

be found at *www.eit.lth.se/goto/kapetanovicprecoders*.

3.4.1 $N = 2$ with QPSK inputs

As already mentioned in Section 3.2.4, this optimization problem has been completely solved in [67]. The authors of [67] analytically proved that the optimal precoders $\boldsymbol{F}$ are such that the Gram matrices $\boldsymbol{G}$ only have two different structures, up to a scaling. One of the two structures has rank 1, while the other has full rank. Which of the two to use depends on the particular realization of $\boldsymbol{S}$.

3.4.2 $N = 2$ with 16-QAM inputs

For 16-QAM inputs, the work in [72] suggests that there should be 8 different precoder structures (i.e., 8 different structures of the resulting $\boldsymbol{G}$ matrices, up to scaling). One of these structures is however not optimal; by running the iterative optimization procedure in Section 3.3, we obtain one precoder structure that has a significantly larger minimum distance than one of the 8 found in [72]. 7 of the 8 precoders have full rank while 1 precoder has rank 1.

In terms of symbol error rate (SER), the newly found precoder has only minor impact. For channels $\boldsymbol{H}$ where the channel coefficients are independent zero-mean, unit-variance, complex Gaussian random variables, the channels where the newly found precoder is optimal are scarce so that the effect of the new precoder almost does not show up in simulations. Thus, the improvement of [72] is mainly of theoretical interest, but it shows that the iterative optimization approach is highly efficient.

3.4.3 $N = 3$ with QPSK inputs

By studying the resulting optimal $\boldsymbol{G}$ matrices for each $\boldsymbol{S}$, it turns out that there are only 14 different $\boldsymbol{G}$ matrices for $N = 3$, as opposed to the 8 proposed in [73].

The different precoders that arise can be characterized in 3 different classes: I) There are 5 precoders with full rank, II) 8 precoders with rank 2, III) 1 precoder with rank 1. The rank deficient precoders are used when some channel eigenvalues are small, so only transmission over the stronger eigenmode occurs.

Unlike Section 3.4.2, the newly found codebook of precoders performs slightly better than the codebook proposed in [73]. A simulation is presented in Figure 3.6. In the simulation, the channel coefficients in $\boldsymbol{H}$ are independent, zero-mean, unit-variance, complex Gaussian random variables. We plot the BER for the 14 newly found precoders, the BER for the 8 precoders

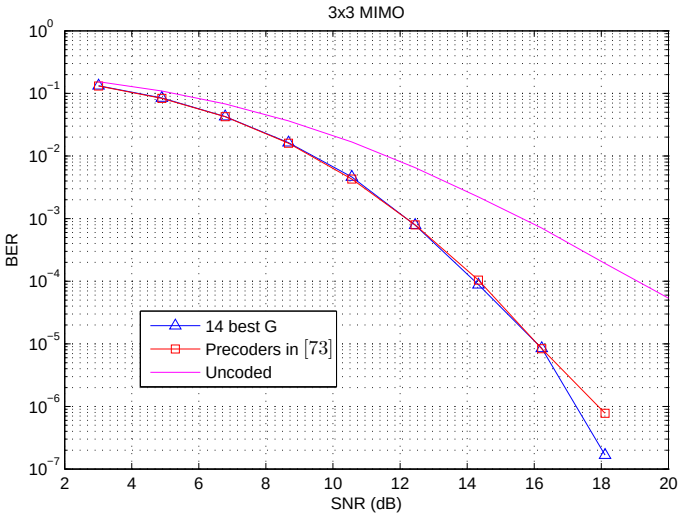

Figure 3.6: BER comparison between the new 14 precoders, the precoders in [73] and uncoded transmission. Since the gain in $D^2_{\min}(\boldsymbol{HF})$ is not substantial, as illustrated in Figure 3.7, the gain in BER is also less pronounced.

from [73] and the BER for uncoded transmission. As can be seen from the figure, there is not much gain compared to the precoders from [73], which suggests that they are close to optimal. However, plotting the distribution of $D^2_{\min}(\boldsymbol{HF})$ for the 14 precoders and the precoders from [73], we see a slight improvement in $D^2_{\min}(\boldsymbol{HF})$ for the 14 precoders. This is illustrated in Figure 3.7.

3.4.4 $N = 4$ with QPSK inputs

In the case of $N = 4$, the number of optimal structures of the $\boldsymbol{G}$ matrices that our iterative optimization was able to identify is 77. With complex Gaussian distributed channels, 30 different precoder structures cover >99.9 % of the channels; hence, the other 47 structures are used very seldomly. Further, in terms of minimum distance, the 30 precoders perform well as there is not much loss compared with using the complete codebook.

Out of the 30 precoders, 8 have full rank, 19 have rank 3, 3 have rank 2, but there is no precoder with rank 1 since in complex Gaussian distributed

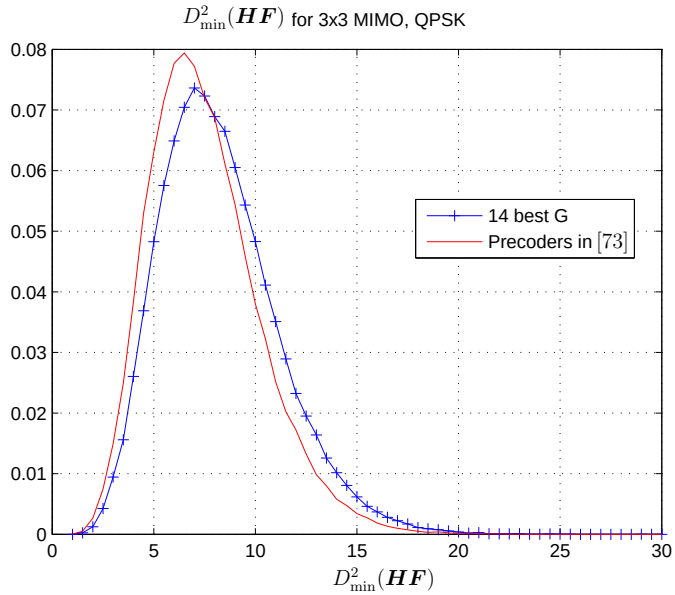

Figure 3.7: Probability function of $D^2_{\min}(\boldsymbol{HF})$ for the 14 best precoders and the precoders in [73]. The precoders in [73] are close to optimal in $D^2_{\min}(\boldsymbol{HF})$ according to the authors – thus we might expect that the new 14 precoders are very close to optimal, if not optimal.

channels, the probability of three very weak eigenmodes is extremely small.

A BER simulation is provided in Figure 3.8. We plot the performance of the 77 suboptimal precoders, a codebook containing only the 30 precoders, uncoded transmission, and the suboptimal precoding method from [65]. As can be seen, there is about 0.8 dB gain of the proposed codebook compared with the competing scheme from [65]. At BER 10^{-3}, there is a 4 dB gain over uncoded transmission. Also, as seen, there is no performance loss by using the 30 best precoders. In Figure 3.9, the distribution of $D^2_{\min}(\boldsymbol{HF})$ is illustrated for the 30 precoders and the precoders in [65]. There is a significant gain in minimum distance for the 30 precoders, which explains the BER gain.

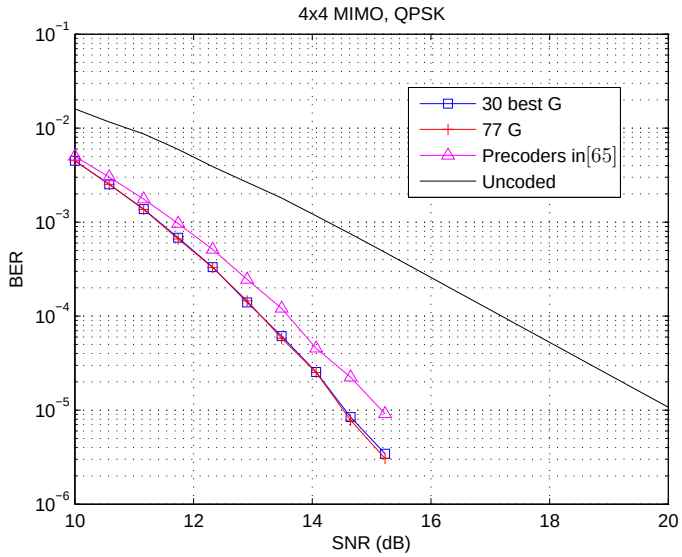

Figure 3.8: BER comparison between the 77 suboptimal $\boldsymbol{G}$, the 30 most occuring $\boldsymbol{G}$ out of the 77, the precoders in [65] and uncoded transmission. There is no loss in using only the 30 most occuring precoders, and the gain in BER is significant compared to the state-of-the art precoders in [65].

3.5 Connection With Lattices

An interesting fact about the $\boldsymbol{G}$ matrices is the structure of the elements. Namely, with proper scaling for each $\boldsymbol{G}$, the elements $g_{j,k} = a_{j,k}+b_{j,k}i$ are either such that $a_{j,k}$ and $b_{j,k}$ are rational numbers, or such that $a_{j,k} = r_1+r_2/\sqrt{2}$ and $b_{j,k} = r_3 + r_4/\sqrt{3}$, where r_1, r_2, r_3, r_4, are rational numbers. A similar ordered structure has also been observed for the $\boldsymbol{G}$ matrices obtained from the optimal precoders in [67, 72]. This hints that there is a hidden underlying structure in the precoding problem. It will be demonstrated that there is indeed a profound relationship between the obtained $\boldsymbol{G}$ matrices and standard lattices. For this reason, we now introduce a brief account on lattice theory.

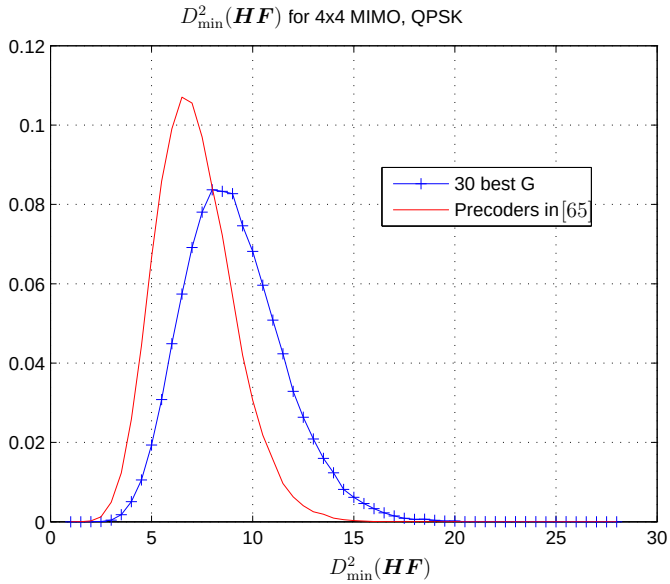

Figure 3.9: Probability function of $D^2_{\min}(\boldsymbol{HF})$ for the 30 new precoders and the precoders in [65]. As seen, there is a significant improvement in $D^2_{\min}(\boldsymbol{HF})$, which carries over to BER as was illustrated in Figure 3.8.

3.5.1 Lattices

All matrices and vectors in this subsection are assumed to be real-valued. This covers complex-valued matrices and vectors too, since any complex-valued matrix $\boldsymbol{A}$ is isomorphic to a real-valued matrix $\boldsymbol{A}_r$ through the transformation

$$\boldsymbol{A}_r = \begin{bmatrix} \mathrm{Re}\{\boldsymbol{A}\} & \mathrm{Im}\{\boldsymbol{A}\} \\ -\mathrm{Im}\{\boldsymbol{A}\} & \mathrm{Re}\{\boldsymbol{A}\} \end{bmatrix} \tag{3.22}$$

and similarly

$$\boldsymbol{s}_r = \begin{bmatrix} \mathrm{Re}\{\boldsymbol{s}\} \\ \mathrm{Im}\{\boldsymbol{s}\} \end{bmatrix} \tag{3.23}$$

for complex-valued vectors $\boldsymbol{s}$, where, $\mathrm{Re}\{\cdot\}$ and $\mathrm{Im}\{\cdot\}$ denote the real and imaginary parts of a matrix/vector, respectively.

Let $\boldsymbol{L} \in \mathbb{R}^{N\times N}$ and let the columns of $\boldsymbol{L}$ be denoted by $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_N$. A lattice

$\Lambda_{\boldsymbol{L}}$ is the set of points

$$\Lambda_{\boldsymbol{L}} = \{\boldsymbol{L}\boldsymbol{u} \ : \ \boldsymbol{u} \in \mathbb{Z}^N\}. \tag{3.24}$$

In (3.24), $\boldsymbol{u}$ is an integer vector and $\boldsymbol{L}$ is called a *generator matrix* for the lattice $\Lambda_{\boldsymbol{L}}$. The *squared minimum distance* of $\Lambda_{\boldsymbol{L}}$ is defined as:

$$D^2_{\min}(\boldsymbol{L}) = \min_{\boldsymbol{u}\neq\boldsymbol{v}} \|\boldsymbol{L}(\boldsymbol{u}-\boldsymbol{v})\|^2 = \min_{\boldsymbol{e}\neq\boldsymbol{0}_N} \|\boldsymbol{L}\boldsymbol{e}\|^2 = \min_{\boldsymbol{e}\neq\boldsymbol{0}_N} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}}\boldsymbol{e},$$

where $\boldsymbol{u}, \boldsymbol{v}$ and $\boldsymbol{e} = \boldsymbol{u}-\boldsymbol{v}$ are integer vectors and $\boldsymbol{G}_{\boldsymbol{L}}$ is the Gram matrix for the lattice $\Lambda_{\boldsymbol{L}}$. The *fundamental volume* is $\mathrm{Vol}(\Lambda_{\boldsymbol{L}}) = |\det(\boldsymbol{L})|$, i.e., it is the volume spanned by $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_N$. Let $\boldsymbol{p}_j$ denote a lattice point in $\Lambda_{\boldsymbol{L}}$. A Voronoi region around a lattice point $\boldsymbol{p}_j$ is the set $\mathcal{V}_{\boldsymbol{p}_j}(\Lambda_{\boldsymbol{L}}) = \{\boldsymbol{w} \ : \ \|\boldsymbol{w}-\boldsymbol{p}_j\| \leq \|\boldsymbol{p}_k-\boldsymbol{w}\|, \ \boldsymbol{p}_k \in \Lambda_{\boldsymbol{L}}\}$. Due to the symmetry of a lattice, it holds that $\mathcal{V}_{\boldsymbol{p}_j}(\Lambda_{\boldsymbol{L}}) = \boldsymbol{p}_j + \mathcal{V}_{\boldsymbol{0}_N}(\Lambda_{\boldsymbol{L}})$. The Voronoi region around $\boldsymbol{0}_N$ is denoted $\mathcal{V}(\Lambda_{\boldsymbol{L}})$.

As can be seen from the definition of $\Lambda_{\boldsymbol{L}}$, the column vectors $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_N$ form a *basis* for the lattice. There are infinitely many bases for a lattice. Assume that $\boldsymbol{L}'$ is another basis for $\Lambda_{\boldsymbol{L}}$. It holds that $\boldsymbol{L}' = \boldsymbol{L}\boldsymbol{Z}$, where $\boldsymbol{Z}$ is a unimodular matrix, i.e., $\boldsymbol{Z}$ has integer entries and $\det(\boldsymbol{Z}) = \pm 1$ [74]. Hence, the generator matrix $\boldsymbol{L}'$ generates the same lattice as $\boldsymbol{L}$, i.e., $\Lambda_{\boldsymbol{L}} \equiv \Lambda_{\boldsymbol{L}'}$ where $\equiv$ denotes equality between sets. Two Gram matrices $\boldsymbol{G}_{\boldsymbol{L}_1} = \boldsymbol{L}_1^{\mathrm{T}}\boldsymbol{L}_1$ and $\boldsymbol{G}_{\boldsymbol{L}_2} = \boldsymbol{L}_2^{\mathrm{T}}\boldsymbol{L}_2$ are *isometric* if there exists a unimodular $\boldsymbol{Z}$ and a constant c such that $\boldsymbol{G}_{\boldsymbol{L}_1} = c\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}_2}\boldsymbol{Z}$. Geometrically, this means that $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ are the same lattice up to rotation and scaling of the basis vectors.

From the definition of the different lattice measures, it follows that

$$D^2_{\min}(\boldsymbol{W}\boldsymbol{L}\boldsymbol{Z}) = D^2_{\min}(\boldsymbol{L}) \tag{3.25}$$

where $\boldsymbol{W}$ is any orthogonal matrix. Similarly, $\mathrm{Vol}(\Lambda_{\boldsymbol{W}\boldsymbol{L}\boldsymbol{Z}}) = \mathrm{Vol}(\Lambda_{\boldsymbol{L}})$.

A number of lattices are especially interesting and have been given formal names in the literature. In particular, the densest lattices in the sense that they maximize the quotient $D^2_{\min}(\Lambda)/\mathrm{Vol}(\Lambda)$ are of interest. In 2, 4, 6, and 8 dimensions, the densest lattices are the Hexagonal A_2, Schläfli D_4, E_6, and the Gosset E_8 lattices, respectively [74]. Apart from these 4, we will also make use of the 2-dimensional square lattice Z_2 and the 6-dimensional D_6 lattice.

3.5.2 Lattice identification of full rank precoders

The optimal precoder structures from Section 3.4 are optimized in terms of their Gram matrices, thus, we do not directly obtain the lattice generators. Therefore, we need to recover the lattice generator matrix from its Gram matrix [75, 76]. However, our task will be easier since we shall first guess the lattice generator, and then verify whether the guess is correct.

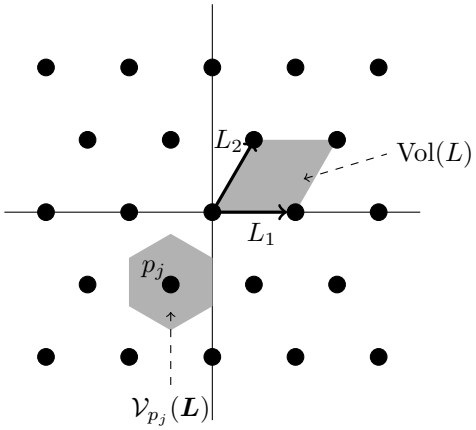

Figure 3.10: The hexagonal lattice depicted with a geometrical description of the introduced lattice quantities.

We next turn the attention to such a verification method. Let $\boldsymbol{G}_r$ denote the real-valued representation of the (complex-valued) Gram matrix $\boldsymbol{G}$ obtained from the iterative optimization described in Section 3.3. The dimension of $\boldsymbol{G}_r$ is $2N \times 2N$. Let $\Lambda_{\boldsymbol{L}_r}$ denote the lattice corresponding to $\boldsymbol{G}_r$. There is yet no explicit information about this lattice. We now make a guess on which lattice $\Lambda_{\boldsymbol{L}_t}$ that $\boldsymbol{G}_r$ corresponds to. Since minimum distance precoding is related to sphere packing problems, we shall later make the "educated guess" that $\Lambda_{\boldsymbol{L}_t}$ is the densest sphere packing lattice. However, this guess will most interestingly not always be correct! Let $\boldsymbol{G}_t = \boldsymbol{L}_t^{\mathrm{T}} \boldsymbol{L}_t$ denote the real-valued Gram matrix of the lattice generator $\boldsymbol{L}_t$ for $\Lambda_{\boldsymbol{L}_t}$. Further, we scale $\boldsymbol{G}_t$ so that

$$\det(\boldsymbol{G}_r) = \det(\boldsymbol{G}_t).$$

Our task is now to verify whether it is true that $\Lambda_{\boldsymbol{L}_r} \equiv \Lambda_{\boldsymbol{L}_t}$.

From Section 3.5.1, it follows that if $\Lambda_{\boldsymbol{L}_r} \equiv \Lambda_{\boldsymbol{L}_t}$ then it must hold that $\boldsymbol{L}_r = \boldsymbol{U}\boldsymbol{L}_t\boldsymbol{Z}$, for some unitary matrix $\boldsymbol{U}$ and where $\boldsymbol{Z}$ is a unimodular matrix. This gives

$$\begin{aligned} \boldsymbol{G}_r &= \boldsymbol{L}_r^{\mathrm{T}} \boldsymbol{L} \\ &= \boldsymbol{Z}^{\mathrm{T}} \boldsymbol{L}_t^{\mathrm{T}} \boldsymbol{L}_t \boldsymbol{Z} \\ &= \boldsymbol{Z}^{\mathrm{T}} \boldsymbol{G}_t \boldsymbol{Z}. \end{aligned} \tag{3.26}$$

Thus, if such unimodular $\boldsymbol{Z}$ exists, we know that $\boldsymbol{G}_r$ and $\boldsymbol{G}_t$ represent the

same lattice. To find such $\boldsymbol{Z}$ is not as straightforward as it may seem at a first glance due to the unimodular constraint. Integer relation algorithms, such as the PSLQ algorithm [77], that find an integer vector $\boldsymbol{a} = (a_1, \ldots, a_n)$ solving $a_1 x_1 + \ldots a_n x_n = 0$ for some real/complex-valued numbers $x_1, \ldots, x_n$, cannot be used in this case since the equation in (3.26) is quadratic in the (integer) elements of $\boldsymbol{Z}$. Furthermore, (3.26) describes a system of quadratic integer equations, one for each element in $\boldsymbol{G}_r$, while PSLQ only considers a single linear integer equation. Thus, we embark on developing an efficient algorithm that finds $\boldsymbol{Z}$ satisfying the equation in (3.26).

If such $\boldsymbol{Z}$ exist, we must have that

$$D^2_{\min}(\boldsymbol{L}_r) = D^2_{\min}(\boldsymbol{L}_t).$$

In terms of the Gram matrices, this condition is expressed as

$$\min_{\boldsymbol{b}} \boldsymbol{b}^{\mathrm{T}} \boldsymbol{G}_r \boldsymbol{b} = \min_{\boldsymbol{b}} \boldsymbol{b}^{\mathrm{T}} \boldsymbol{G}_t \boldsymbol{b}. \tag{3.27}$$

If (3.27) holds, consider the set of minimal norm vectors $\Omega_r = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{K_r}\}$ for $\boldsymbol{G}_r$ and $\Omega_t = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{K_t}\}$ for $\boldsymbol{G}_t$. These sets can be found via, e.g., sphere decoding [78]. In order for $\boldsymbol{L}_r$ and $\boldsymbol{L}_t$ to generate the same lattice, it must hold that $K_r = K_t$, i.e., the kissing numbers of the lattices must be the same. If not, the lattices are not the same.

Given that the two Gram matrices $\boldsymbol{G}_r$ and $\boldsymbol{G}_t$ have the same minimum distances and kissing numbers, we can try to construct the unimodular $\boldsymbol{Z}$. Let $\boldsymbol{A}_r = [\boldsymbol{v}^r_{j_1}, \ldots, \boldsymbol{v}^r_{j_{2N}}]$ be a matrix formed by $2N$ linearly independent vectors from Ω_r. Since $\boldsymbol{Z}$ represents a change of basis of the lattice, it must hold that there exists a subset of $2N$ vectors $\boldsymbol{A}_t = [\boldsymbol{v}^t_{k_1}, \ldots, \boldsymbol{v}^t_{k_{2N}}]$ from Ω_t such that $\boldsymbol{A}_r = \boldsymbol{Z}\boldsymbol{A}_t$. Since $\boldsymbol{A}_r$ and $\boldsymbol{A}_t$ are invertible and contains only integer elements by assumption, it follows that $\boldsymbol{A}_t^{-1}\boldsymbol{A}_r$ is always an integer valued-matrix. Using (3.26), we know that if

$$(\boldsymbol{A}_t)^{\mathrm{T}} (\boldsymbol{A}_r^{-1})^{\mathrm{T}} \boldsymbol{G}_r \boldsymbol{A}_r^{-1} \boldsymbol{A}_t = \boldsymbol{G}_t, \tag{3.28}$$

we have found the unimodular matrix as $\boldsymbol{Z} = \boldsymbol{A}_t^{-1}\boldsymbol{A}_r$. Consequently, a brute force approach exhausts all subsets of $\boldsymbol{V}_t$ and checks whether (3.28) is satisfied. If no subset $\boldsymbol{A}_t$, so that (3.28) holds, exists, the guess that $\Lambda_{\boldsymbol{L}_r} \equiv \Lambda_{\boldsymbol{L}_t}$ is incorrect.

To exhaustively test all subsets is inefficient. Instead we have used a recursive branch-and-bound algorithm, which goes as follows. Consider (3.28). The task is to form $\boldsymbol{A}_t$ as $2N$ vectors from Ω_t so that (3.28) holds. Suppose initially that we pick the first column in $\boldsymbol{A}_t$ as $\boldsymbol{a}_1$. It must then hold that

$$\boldsymbol{a}_1^{\mathrm{T}} (\boldsymbol{A}_r^{-1})^{\mathrm{T}} \boldsymbol{G}_r \boldsymbol{A}_r^{-1} \boldsymbol{a}_1 = g_{t,1,1}, \tag{3.29}$$

where $g_{t,1,1}$ denotes element (1,1) of the matrix $\boldsymbol{G}_t$. If this does not hold, $\boldsymbol{a}_1$ cannot possibly be the first column of $\boldsymbol{A}_t$.

Now suppose that (3.29) holds, and suppose that we choose $\boldsymbol{a}_2$ as the second column. Then it must hold that

$$[\boldsymbol{a}_1\boldsymbol{a}_2]^{\mathrm{T}}(\boldsymbol{A}_r^{-1})^{\mathrm{T}}\boldsymbol{G}_r\boldsymbol{A}_r^{-1}[\boldsymbol{a}_1\boldsymbol{a}_2] = g_{t,1:2,1:2}, \tag{3.30}$$

where $g_{t,1:k,1:k}$ denotes the sub-matrix formed by the elements from the first k rows and columns of $\boldsymbol{G}_t$.

If (3.30) does not hold, we do not have to consider the combinations $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ any further. In this fashion, we can branch-and-bound until we have found a solution $\boldsymbol{A}_t$ to (3.28), or until we have exhausted all possibilites without finding any valid solution. The pseudo-code of a recursive implementation of the branch-and-bound algorithm is given in Table 3.2.

With a standard work-station and $N = 4$, the entire verification process is a matter of fractions of a second.

3.5.3 Lattice classification of rank deficient Gram matrices

Whenever the channel $\boldsymbol{S}$ contains at least one small eigenvalue, the optimal precoder structure $\boldsymbol{G}$ is rank deficient. Let β denote the rank of the $2N \times 2N$ matrix $\boldsymbol{G}$ - note that β must be an even integer since the eigenvalues of the real-valued matrix $\boldsymbol{G}_r$ appear in pairs. This implies that we can not hope to identify $\boldsymbol{G}$ as the Gram matrix of any well known $2N$-dimensional lattice, since these are all full rank. However, $\boldsymbol{G}$ can still represent an β-dimensional ordered structure. This ordered structure may, or may not, represent a lattice, as is recalled next.

Let $\boldsymbol{L}_r$ denote any $\beta \times 2N$ matrix such that $\boldsymbol{G}_r = \boldsymbol{L}_t^{\mathrm{T}}\boldsymbol{L}_t$. It can, e.g., be obtained by taking the QR decomposition of $\boldsymbol{L}_r$ and setting $\boldsymbol{L}_t$ to be the upper-triangular matrix of the QR factorization. The generator matrix $\boldsymbol{L}_r$ corresponds to $2N$ vectors in β-space. Through $\Lambda = \{\boldsymbol{L}_r\boldsymbol{v} \,|\, \boldsymbol{v} \in \mathbb{Z}^{2N}\}$ the generator spans a number of points in β-space. These points can be a subset of, or coincide with, the points spanned by an $\beta \times \beta$ lattice generator $\boldsymbol{L}_{\mathrm{sq}}$, the subscript "sq" denotes "square". If so, the implication is that for any integer vector $\boldsymbol{v} \in \mathbb{Z}^{2N}$ there exists a corresponding integer vector $\boldsymbol{u} \in \mathbb{Z}^{\beta}$ such that

$$\boldsymbol{L}_{\mathrm{sq}}\boldsymbol{u} = \boldsymbol{L}_r\boldsymbol{v}.$$

Since this should hold for all integer vectors $\boldsymbol{v}$, it follows that it must be possible to factorize $\boldsymbol{L}_r$ as

$$\boldsymbol{L}_r = \boldsymbol{L}_{\mathrm{sq}}\boldsymbol{P}, \tag{3.31}$$

Table 3.2: Recursive algortihm to find the matrix $\boldsymbol{Z}$. If $\boldsymbol{Z}$ is empty when the routine terminates, the two lattices are not the same. The tolerance ϵ can be taken as any small number, e.g., 10^{-4}.

Algorithm to find the matrix $\boldsymbol{Z}$

$[\boldsymbol{Z}] = \text{FINDZMATRIX}(\boldsymbol{A}_r, \boldsymbol{A}_t, \boldsymbol{G}_r, \boldsymbol{G}_t, \boldsymbol{p})$

INPUTS: $\boldsymbol{A}_r$, $\boldsymbol{A}_t = [\,]$, $\boldsymbol{G}_r$, $\boldsymbol{G}_t$, $\boldsymbol{p} = [1, 2, \ldots, \kappa]$, WHERE κ IS THE KISSING NUMBER OF $\boldsymbol{G}_r$.

OUTPUT: THE $\boldsymbol{Z}$ MATRIX IN (3.26) IF IT EXISTS.

```
If number of columns in A_t = 2N
        Z = A_t A_r^{-1}
else
        T = A_r^T G_r A_r
        Z = 0
        For i ∈ p
                e = (A_r)_{1:end,i}
                A_t = [A_t e]
                C = A_t^T G_t A_t
                m = number of rows in C
                if ||T_{1:m,1:m} − C||^2 < ε then
                   q = [p_1, ..., p_{i−1}, p_i, ... p_end]
                   [Z] = FINDZMATRIX(A_r, A_t, G_r, G_t, q)
                end
                if Z is empty
                        break
                end
        end
end
```

where $\boldsymbol{P}$ is an $\beta \times 2N$ integer matrix. If such factorization exist, the lattice $\Lambda_{\boldsymbol{L}_r}$ is a subset of the lattice $\Lambda_{\boldsymbol{L}_{\text{sq}}} = \{\boldsymbol{L}_{\text{sq}}\boldsymbol{u} \,|\, \boldsymbol{u} \in \mathbb{Z}^\beta\}$. Let $\boldsymbol{l}_{r,j_1}, \ldots, \boldsymbol{l}_{r,j_\beta}$ be β linearly independent columns in $\boldsymbol{L}_r$ (they certainly exist since $\boldsymbol{L}_r$ has rank β). From (3.31), it follows that

$$[\boldsymbol{l}_{r,j_1}, \ldots, \boldsymbol{l}_{r,j_\beta}] = \boldsymbol{L}_{\text{sq}} \underbrace{[\boldsymbol{p}_{j_1}, \ldots, \boldsymbol{p}_{j_\beta}]}_{\boldsymbol{P}_{j_1:j_\beta}},$$

where $\boldsymbol{P}_{j_1}, \ldots, \boldsymbol{P}_{j_\beta}$ are β independent columns of $\boldsymbol{P}$. Since $\boldsymbol{P}_{j_1:j_\beta}$ is a non-degenerate integer matrix, it holds that any $\beta \times 1$ integer vector can be expressed as $\boldsymbol{P}_{j_1:j_\beta}\boldsymbol{r}$, where $\boldsymbol{r}$ is a $\beta \times 1$ vector with rational entries. Hence, this directly implies that every column in $\boldsymbol{L}_r$ can be expressed as a rational combination of $\boldsymbol{l}_{r,j_1}, \ldots, \boldsymbol{l}_{r,j_\beta}$. If this is not possible, then $\boldsymbol{L}_r$ does not represent a lattice. If $\boldsymbol{P}$ contains the identity matrix, we in fact have that $\Lambda_{\boldsymbol{L}_r} = \Lambda_{\boldsymbol{L}_{\text{sq}}}$. However, if the identity is not contained in $\boldsymbol{P}$, there is ambiguity in the factorization $\boldsymbol{L}_r = \boldsymbol{L}_{\text{sq}}\boldsymbol{P}$ since many integer matrices $\boldsymbol{P}$ may exist. This can be resolved by considering the factorization with the largest value of $\text{Vol}(\boldsymbol{L}_{\text{sq}})$. However, we will only consider factorizations where $\boldsymbol{P}$ contains the identity matrix, so that $\Lambda_{\boldsymbol{L}_r} = \Lambda_{\boldsymbol{L}_{\text{sq}}}$. For the cases where no such factorization exists, we can still identify well known sub-lattices in $\Lambda_{\boldsymbol{L}_r}$. Suppose that a subset of the columns in $\boldsymbol{L}_r$ are equivalent to some lattice Λ_1 and that the remaining columns are equivalent to another lattice Λ_2. We write this as $\Lambda_{\boldsymbol{L}_r} \equiv \Lambda_1 \times \Lambda_2$. The subsets of columns in $\boldsymbol{L}_r$ that form the lattices Λ_1 and Λ_2 may be rotated by unitary transforms $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$, respectively, and also scaled by constants c_1 and c_2.

3.5.4 Lattice classification of the optimal precoders

In this section we examine the Gram matrices of the optimized precoders from Section 3.4. It will turn out that all full rank precoders can be classified as instances of well known lattices. Interestingly, in 6 dimensions this lattice is the D_6 lattice and not the denser E_6 lattice. For the rank deficient precoders, some are instances of lattices, while others are built up from stacking several lower dimensional lattices. We have not found a precoder that does not correspond to a well known lattice.

$N = 2$ with QPSK inputs

The structures of all precoders are listed in Table 3.3. As explained in Section 3.4.1, only two different precoder structures $\boldsymbol{G}_r$ occur for $N = 2$ with QPSK inputs. One has full rank 4 and one has rank 2.[5] Using our lattice identification

[5]The reader is reminded that $\boldsymbol{G}_r$ is the real-valued representation of $\boldsymbol{G}$. The full rank of $\boldsymbol{G}$ is 2 while it is 4 for $\boldsymbol{G}_r$.

method from Sections 3.5.2 and 3.5.3, we are able to characterize their corresponding lattices. For the full rank precoder, $\boldsymbol{G}_r$ is identified as the Schläfli lattice D_4; this is also the densest lattice in 4 dimensions. For the rank deficient precoder, no factorization (3.31) exists where $\boldsymbol{P}$ contains the identity matrix. This can be seen as follows. With suitable scaling, $\boldsymbol{G}_r$ can be expressed as $\boldsymbol{G}_r = \boldsymbol{L}_r^{\mathrm{T}} \boldsymbol{L}_r$ with

$$\boldsymbol{L}_r = \begin{bmatrix} 4+2\sqrt{3} & -1 & 0 & 2+\sqrt{3} \\ 0 & 2+\sqrt{3} & -4-2\sqrt{3} & 1 \end{bmatrix}.$$

The middle two columns of this $\boldsymbol{L}_r$ are linearly independent, so the other two columns in $\boldsymbol{L}_r$ must be a rational combination of these if $\boldsymbol{L}_r$ is to represent a lattice. However, it is obvious that this is not the case, since the other two columns contain the irrational number $\sqrt{3}$ as its first coordinate. Nevertheless, we can observe that the precoder contains two sub-lattices that both are instances of the Z_2 lattice, namely column 1 and 3, and column 2 and 4, respectively. Hence, the precoder structure is $Z_2 \times Z_2$. These two sub-lattices are rotated against each other, but are also differently power scaled. For an analytical derivation of this precoder, see [67].

$N = 2$ with 16-QAM inputs

The structures of all precoders are listed in Table 3.4. There are 8 different precoder structures presented in [72]; 7 of them have full rank, while one only has rank 2. One of the 7 full rank precoders is however not optimal. This was realized in the following way. Out of the 7 full rank precoders, 6 can be identified as the Schläfli lattice. The 7th precoder is "close" to the Schläfli lattice, but there is no exact match. Therefore, it is suspected that this precoder could therefore be incorrect and we applied the iterative optimization technique from Section 3.3 for the channel matrices $\boldsymbol{S}$ where the non-Schläfli precoder was to be used. The iterative optimization quickly produces a better precoder, which can be identified as the Schläfli lattice.

All of the 7 full rank precoders can be identified as the Schläfli lattice, while the rank 2 precoder has a structure of the form $Z_2 \times Z_2$.

$N = 3$ with QPSK inputs

The structure of all precoders are listed in Table 3.5. Out of the 14 optimal $\boldsymbol{G}_r$ that were found for this case, eight have full rank 6, five have rank 4 and one has rank 2. The precoders with full rank are identified as the D_6 lattice. Very interestingly, this lattice is not the densest lattice in 6 dimensions. The densest lattice, E_6, does in fact never show up! In [79], sub-optimal lattice

based precoders are constructed that make use of the densest lattices. Our observation shows that also other lattices should be considered. Although it cannot be guaranteed that the iterative optimization converges to the optimum, it nevertheless produces solutions that hint upon a structure which is believed to be optimal. Thus, there is room for improvement of the results in [79], and this will be investigated more closely in Chapter 4.

The 5 precoders with rank 4 can be identified as the Schläfli lattice in 4 dimensions. As an example, we study one of these 5 precoders in closer detail. Its Gram matrix can be factored as $\boldsymbol{L}_r^{\mathrm{T}}\boldsymbol{L}_r$ with

$$\boldsymbol{L}_r = \begin{bmatrix} 1 & 0 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \sqrt{2} & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{\sqrt{3}}{2} & -\frac{1}{\sqrt{3}} & -(1+\frac{1}{\sqrt{3}}) & \frac{1}{2\sqrt{3}} \\ 0 & 0 & 0 & \frac{2}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \end{bmatrix}.$$

Although this matrix contains irrational numbers, these numbers appear in a way so that a factorization of the form (3.31) is still possible. If we construct $\boldsymbol{L}_{\mathrm{sq}}$ as columns 1, 3, 4, and 6 from $\boldsymbol{L}_r$, we get $\boldsymbol{L}_r = \boldsymbol{L}_{\mathrm{sq}}\boldsymbol{P}$ with,

$$\boldsymbol{P} = \begin{bmatrix} 1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -2 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Since $\boldsymbol{P}$ only has integer elements and contains the identity matrix, it follows that $\boldsymbol{L}_r$ and $\boldsymbol{L}_{\mathrm{sq}}$ span the same lattice. Then the lattice spanned by $\boldsymbol{L}_{\mathrm{sq}}$ can be identified by the method described in Section 3.5.2, which yields the Schläfli lattice with

$$\boldsymbol{Z} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Finally, the rank 2 precoder has a structure of the form $Z_2 \times Z_2 \times Z_2$.

$N = 4$ with QPSK inputs

The structure of all precoders are listed in Table 3.6. The 8 full rank precoders are all identified as the Gosset lattice E_8. Although one cannot make a certain claim of the total size of the optimal precoder codebook, there is evidence to believe that the optimal codebook only contains 8 full rank precoders.

2 of the 19 rank 6 precoders are the D_6 lattice, 15 have the structure $D_6 \times Z_2$, and 2 have the structure $D_4 \times Z_2 \times Z_2$. The 3 rank 4 precoders are

Table 3.3: Optimal precoders for $N = 2$ with QPSK inputs. Real-valued representation of the precoders.

$N = 2$, QPSK inputs

Rank β	**number of precoders**	**Lattice classification**
4	1	D_4
2	1	$Z_2 \times Z_2$
Total:	2	

Table 3.4: Optimal precoders for $N = 2$ with 16-QAM inputs. Real-valued representation of the precoders.

$N = 2$, 16-QAM inputs

Rank β	**number of precoders**	**Lattice classification**
4	7	D_4
2	1	$Z_2 \times Z_2$
Total:	8	

identified as the D_4 lattice. The proposed codebook for Gaussian channels does not contain any rank 2 precoder. For completeness we have also analyzed the optimal precoder to use for channels with $s_{2,2} = s_{3,3} = s_{4,4} = 0$, i.e, all 4 data streams are multiplexed onto a single eigenmode. It turns out that the optimal precoder has the structure $Z_2 \times Z_2 \times Z_2 \times Z_2$.

Table 3.5: Optimal precoders for $N = 3$ with QPSK inputs. Real-valued representation of the precoders.

$N = 3$, QPSK inputs

Rank β	**number of precoders**	**Lattice classification**
6	5	D_6
4	8	D_4
2	1	$Z_2 \times Z_2 \times Z_2$
Total:	14	

Table 3.6: Optimal precoders for $N = 4$ with QPSK inputs. Real-valued representation of the precoders. * This precoder is not used in the proposed codebook for complex Gaussian channels, but is added to this table for completeness.

$N = 4$, QPSK inputs

Rank β	**number of precoders**	**Lattice classification**
8	8	E_8
6	2	D_6
6	15	$D_6 \times Z_3$
6	2	$D_4 \times Z_2 \times Z_2$
4	3	D_4
2	1^*	$Z_2 \times Z_2 \times Z_2 \times Z_2$
Total:	$30/31^*$	

Chapter 4

Precoding From a Lattice Point of View

The previous chapter presented numerical methods to construct precoders that produce large minimum distances at the receiver. The first method was a sub-optimal explicit construction of the precoders, based on imposing a Toeplitz structure on the Gram matrix. Next, this constraint was dropped, and an iterative optimization algorithm was used that produced precoders improving upon the previous best ones reported in the literature, and believed to be optimal. It was observed that the obtained precoders exhibit a structure in their Gram matrix, which is connected with well-known lattice structures. More precisely, these precoders give rise to received signaling points that are structured as well-known lattices. Taken together with the fact that the algorithm converged to precoders that improve upon previous results, it is believed that they indeed might be optimal. This chapter will thus explore the connection between the minimum distance precoders and lattice theory. Hence, the minimum distance problem will be viewed from a lattice theoretic perspective, and this will enable us to explain the observed structures.

4.1 Introduction

In order to study the minimum distance problem from a lattice point of view, the alphabet $\mathcal{A}$ has to be infinite, i.e., $\boldsymbol{a} \in \mathcal{A}^B = \mathbb{Z}^B[i]$, the set of B-dimensional Gaussian integer vectors. Hence, the error vectors $\boldsymbol{e}$ are B dimensional Gaussian integer vectors. From now on, $\mathcal{A}$ will not be explicitly written out, since it is implicit that it is equal to $\mathbb{Z}[i]$. Thus $D^2_{\min}(\boldsymbol{SF}, \mathcal{A})$ will be denoted as

$D^2_{\min}(\boldsymbol{SF})$. Since there are infinitely many error vectors $\boldsymbol{e}$, the $B \times B$ Gram matrix $\boldsymbol{G}$ must have rank B in order for the inequalities $\boldsymbol{e}^*\boldsymbol{G}\boldsymbol{e} \geq 1$ to hold. If not, then the minimum distance $D^2_{\min}(\boldsymbol{SF}) = \boldsymbol{e}^*\boldsymbol{G}\boldsymbol{e}$ is arbitrarily close to 0, since $\boldsymbol{e}$ can be arbitrarily close to the eigenvectors that corresponds to zero eigenvalues. Thus, $\boldsymbol{G}$ is a positive definite matrix, and $N \geq B$ must hold (the reader is reminded that the precoder $\boldsymbol{F}$ in (3.5) has dimensions $N \times B$, and $\boldsymbol{G} = \boldsymbol{F}^*\boldsymbol{S}^2\boldsymbol{F}$). This chapter only focuses on the case $N = B$, with the remark that the analysis also applies to $N > B$.

Using notions from Section (3.5.1), we start by reformulating (3.6) as a lattice problem. Let $\boldsymbol{M} = \boldsymbol{SF}$ be the lattice generator matrix at the receiver, which as described above, must have full rank. $\boldsymbol{M}$ can be factorized as $\boldsymbol{M} = \boldsymbol{WBZ}$, where $\boldsymbol{W}$ is a unitary/orthogonal matrix, $\boldsymbol{B}$ is a $N \times N$ matrix and $\boldsymbol{Z}$ is a unimodular matrix. The lattice structure of $\boldsymbol{M}$ is determined by the matrix $\boldsymbol{B}$, while $\boldsymbol{Z}$ is the basis through which the lattice is represented. The matrix $\boldsymbol{W}$ is merely a rotation of the lattice, but plays an important role in the optimization to follow. With this factorization of $\boldsymbol{M}$, it follows that $\boldsymbol{F}$ can be written as

$$\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{M} = \boldsymbol{S}^{-1}\boldsymbol{WBZ}. \tag{4.1}$$

Hence, (3.6) can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{B},\boldsymbol{Z}} \operatorname{tr}(\boldsymbol{Z}^*\boldsymbol{B}^*\boldsymbol{W}^*\boldsymbol{S}^{-2}\boldsymbol{WBZ}) \\ &\text{subject to } D^2_{\min}(\boldsymbol{WBZ}) = 1. \end{aligned} \tag{4.2}$$

For completeness, we shall separate between two cases: (i) Real-valued precoding, where all quantities in (4.2) are real-valued, and (ii) Complex-valued precoding, where all quantities, except $\boldsymbol{S}$, are complex-valued.

From Theorem 7 it follows that the optimization over $\boldsymbol{W}$ is straightforward once $\boldsymbol{BZ}$ is fixed: The optimal $\boldsymbol{W}$ equals the left unitary matrix of $\boldsymbol{BZ}$. This leaves us with the optimization of $\boldsymbol{B}$ and $\boldsymbol{Z}$, and we shall start with $\boldsymbol{B}$ in Section 4.2.1, while optimization over $\boldsymbol{Z}$ is treated in Section 4.2.2.

4.2 Optimal Two Dimensional Lattice Precoders

As a start, two dimensional MIMO systems are studied, i.e., $N = 2$. In [79] and [89], it is proposed to design $\boldsymbol{F}$ based on dense lattice packings. A lattice-based construction implicitly assumes that the signal constellation is a finite but sufficiently "large" set of lattice points, and the idea is that if the received constellation points $\boldsymbol{SFa}$'s are arranged as a dense lattice packing, the minimum distance is expected to be "good". However, no exact results on optimality have been presented in either of these papers.

To gain some insight into the problem, let us examine some simple special cases. First, we rewrite (4.2) in its equivalent form

$$\begin{aligned} &\max_{\boldsymbol{F}} D^2_{\min}(\boldsymbol{SF}) \\ &\text{subject to } \operatorname{tr}(\boldsymbol{FF}^*) \le P_0. \end{aligned} \tag{4.3}$$

The formulation in (4.3) turns out to be easier to analyize numerically. In real-valued precoding, some specific instances of the problem in (4.3) can be viewed geometrically. Assume that $\operatorname{tr}(\boldsymbol{FF^*}) = 4$ and the elements of the input $\boldsymbol{a}$ are identically and independently distributed (i.i.d.) random variables. Normalize $\boldsymbol{S}$ to have $s_{2,2} = 1$, which only scales the optimal solution to (4.3) with a constant, so that changing $\boldsymbol{S}$ corresponds to varying the value of $s_{1,1}$. Since there are only four real-valued elements in $\boldsymbol{F}$, and they are bounded by the energy constraint, it is possible to determine the optimal $\boldsymbol{F}$ to (4.3) for some carefully chosen value of $s_{1,1}$, say, by empirical means. When $\boldsymbol{S} = \boldsymbol{I}$ (i.e., $s_{1,1} = 1$), one optimal solution to (4.3) is $\boldsymbol{F} = \boldsymbol{I}$, while another one is

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0.5 \\ 0 & \sqrt{3/4} \end{pmatrix},$$

which spans a hexagonal lattice. However, as soon as $\boldsymbol{S}$ deviates from $\boldsymbol{I}$ (even with a very small change, say, $s_{1,1} = 1.01$), the optimal $\boldsymbol{F}$ is unique (up to sign changes in the columns) and it gives rise to an $\boldsymbol{SF}$ that is a generator matrix for the hexagonal lattice. Varying $s_{1,1}$ further, the optimal $\boldsymbol{F}$ changes in a continuous way, while the received lattice $\boldsymbol{SF}$ remains the same (up to scaling). This behavior continues until $s_{1,1}$ reaches a certain value, for which the optimal $\boldsymbol{F}$ suddenly changes in a discontinuous way, resulting in a discontinuous change in $\boldsymbol{SF}$. However, surprisingly, $\boldsymbol{SF}$ still spans a hexagonal lattice, in spite of its subtle changes!

Figure 4.1 depicts such a behavior by plotting as vectors the columns of the optimal $\boldsymbol{F}$ and the corresponding $\boldsymbol{SF}$ for three different $\boldsymbol{S}$ with $s_{1,1} = 1.5, 2.7$ and 2.8, respectively. The received constellation points $\boldsymbol{SFa}$ are shown as discrete points. The optimal $\boldsymbol{F}$ changes continuously as $s_{1,1}$ increases from 1.5 to 2.7, and the columns of $\boldsymbol{SF}$ are simply being scaled and always span the same hexagonal lattice (up to scaling). When $s_{1,1}$ further increases from 2.7 to 2.8, there is a discontinuous change in the elements of the optimal $\boldsymbol{F}$. The columns of $\boldsymbol{SF}$ also change discontinuously, but they still span the hexagonal lattice (up to scaling and rotation). This intriguing behavior of the optimal precoder poses a challenging puzzle, and the aim of this section is to resolve this puzzle.

Although the results are derived for infinite constellations, by using lattice theory, the results are applicable to "large" QAM constellations. In the nu-

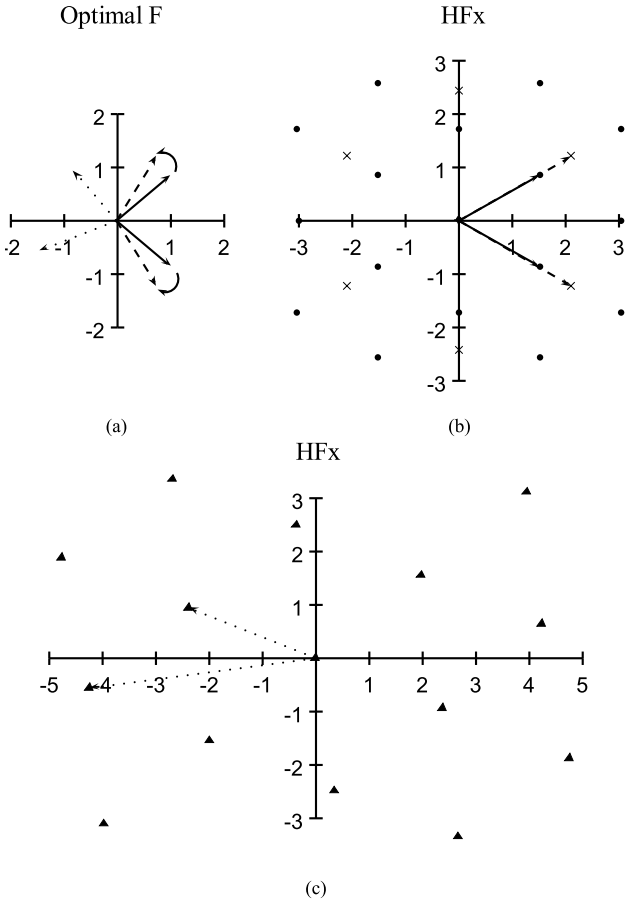

Figure 4.1: Visualization of the solution to the precoding optimization problem in (4.3) when $\mathcal{E}$ is $\mathbb{Z}$, $\mathrm{tr}(\boldsymbol{FF'}) = 4$ and $\boldsymbol{H}$ is a diagonal channel matrix $\boldsymbol{S} = \mathrm{diag}([s_{1,1}\ 1])$. (a) Columns of the optimal real-valued precoding matrix $\boldsymbol{F}$ are plotted. Three different $\boldsymbol{S}$ are considered: $s_{1,1} = 1.5$ (solid line arrow); $s_{1,1} = 2.7$ (dashed line arrow); $s_{1,1} = 2.8$ (dotted line arrow). Columns of the same matrix are plotted as arrows with the same line style. (b) Columns of the matrices $\boldsymbol{SF}$ and their corresponding received constellation points $\boldsymbol{SFa}$'s for $h_{1,1} = 1.5$ (solid line arrows, filled circles) and for $s_{1,1} = 2.7$ (dashed line arrows, crosses) are plotted. (c) Columns of the matrices $\boldsymbol{SF}$ and their corresponding received constellation points $\boldsymbol{SFa}$ for $s_{1,1} = 2.8$ (dotted line arrows, filled triangles) are plotted.

merical result section, we shall investigate how "large" a QAM constellation is sufficient for the presented results to be fruitfully applied. With the solution at hand, we are able to answer questions, such as the following.

- Is there a general underlying structure of the precoding optimization problem (4.2)?
- Under what conditions, does the solution to (4.2) vary with the channel matrix $\boldsymbol{S}$ in a continuous (respectively, discrete) manner?
- Is it possible to offline construct a codebook of optimal precoders so that there is no need to perform any online optimization?

The answers to these questions are that there is indeed a profound structure in the solution of (4.2). Remarkably, there is a single precoder structure which is optimal, and it organizes the received constellation points as a hexagonal lattice for real-valued $\boldsymbol{F}$'s, and as a Schläfli lattice for complex-valued $\boldsymbol{F}$'s. However, the basis through which the lattice $\boldsymbol{SF}$ is observed changes (up to scaling) in a discrete fashion when $\boldsymbol{S}$ changes. This implies that (4.2) is actually a discrete optimization problem and not a continuous one.

4.2.1 Optimal precoding lattices

In this section, the optimal lattice $\boldsymbol{B}$ for the real-valued and the complex-valued cases is derived.

For the real-valued case, the main result is:

Theorem 8. *For any non-singular channel matrix $\boldsymbol{S}$, the optimal lattice $\boldsymbol{B}$ in (4.2) is the hexagonal lattice, i.e.,*

$$\boldsymbol{B} = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}.$$

Proof. First, the constraint in (4.2) will be made more managable. It follows from (3.25) that $D^2_{\min}(\boldsymbol{WBZ}) = D^2_{\min}(\boldsymbol{B})$. Let $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^2$ be the columns of $\boldsymbol{B}$ and assume that $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$. In 1801, C.F. Gauss noted [80] that if $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ fulfill, $|\boldsymbol{b}_2 \cdot \boldsymbol{b}_1| \leq \|\boldsymbol{b}_1\|^2/2$, where "$\cdot$" is the scalar product between vectors, then $D^2_{\min}(\boldsymbol{B}) = \|\boldsymbol{b}_1\|^2$. Given $\boldsymbol{b}_1$, the set of all $\boldsymbol{b}_2$ satisfying the inequality is the *minimum distance region* of $\boldsymbol{b}_1$. Figure 4.2 depicts this region geometrically. $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are actually the *shortest basis* for the lattice, since $\|\boldsymbol{b}_1\|$ is the length of the shortest vector in the lattice, and it can be shown that $\|\boldsymbol{b}_2\|$ is the length of the next shortest vector in the lattice. Hence, by putting $\|\boldsymbol{b}_1\| = 1$ and letting $\boldsymbol{b}_2$ be any vector in the minimum distance region of $\boldsymbol{b}_1$,

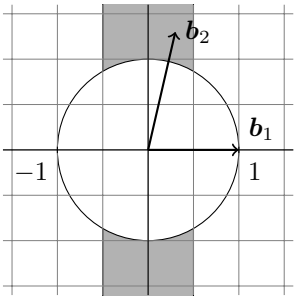

Figure 4.2: The minimum distance region of $\boldsymbol{b}_1$ is shaded. All $\boldsymbol{b}_2$ inside the shaded region generate a lattice, spanned by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$, with a minimum distance equal to the length of $\boldsymbol{b}_1$.

the matrix $\boldsymbol{B}$ will be a generator matrix for any lattice in the plane with unit minimum distance.

Let $r = \|\boldsymbol{b}_2\|$. The constraint $D^2_{\min}(\boldsymbol{WBZ}) = 1$ can be written as $r \geq 1$ and $|\cos(\phi)| \leq 1/2r$ where ϕ is the angle between $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. Hence, $\boldsymbol{WB}$ can be written as

$$\boldsymbol{WB} = \begin{pmatrix} \sin(\alpha) & r\sin(\alpha \pm \phi) \\ \cos(\alpha) & r\cos(\alpha \pm \phi) \end{pmatrix}. \tag{4.4}$$

The optimization (4.2) can now be formulated over α, ϕ and r:

$$\min_{\alpha,\phi,r} \operatorname{tr}(\boldsymbol{Z}^*\boldsymbol{B}^*\boldsymbol{W}^*\boldsymbol{S}^{-2}\boldsymbol{WBZ}) \quad \text{subject to} \quad r \geq 1,\ |\cos(\phi)| \leq 1/2r. \tag{4.5}$$

It follows that the intervals for α and ϕ are $0 \leq \alpha \leq 2\pi$, $|\phi| \leq \cos^{-1}(1/2r)$.

Let $s_{1,1}, s_{2,2}$ be the diagonal elements of $\boldsymbol{S}$ and assume $s_{1,1} \geq s_{2,2}$. For notational convenience, we let z_{jj} be the elements in $\boldsymbol{Z}$. Define $s \triangleq s_{2,2}/s_{1,1}$ and

$$a = \frac{z_{11}^2 + z_{12}^2}{\|\boldsymbol{Z}\|^2} \quad b = \frac{z_{11}z_{21} + z_{12}z_{22}}{\|\boldsymbol{Z}\|^2} \quad c = \frac{1+s^2}{2}. \tag{4.6}$$

In order to obtain easier expressions, we scale the objective function (4.5) with $1/s_{2,2}\|\boldsymbol{Z}\|^2$ which has no impact on the solution, and by doing so we get the following objective function

$$\begin{aligned} f(\alpha,\phi,r) &\triangleq \operatorname{tr}(\boldsymbol{Z}^*\boldsymbol{B}^*\boldsymbol{W}^*\boldsymbol{S}^{-2}\boldsymbol{WBZ})/s_{2,2}\|\boldsymbol{Z}\|^2 \\ &= c(a + r^2(1-a) + 2br\cos(\phi)) \\ &\quad +(1-c)(a\cos(2\alpha) + (1-a)r^2\cos(2\alpha+2\phi) + 2br\cos(2\alpha+\phi)). \end{aligned} \tag{4.7}$$

Since $0 \leq s \leq 1$, it follows that $1/2 \leq c \leq 1$.

First, we minimize $f(\alpha, \phi, r)$ over α by making use of the following Lemma

Lemma 1. *Let $g(x) = \sum_{j=1}^{n} a_j \cos(x+\theta_j)$ for some real-valued constants $\{a_j\}$ and $\{\theta_j\}$. It holds that*

$$\min_x g(x) = -\sqrt{\sum_{j=1,k=1}^{n} a_j a_k \cos(\theta_j - \theta_k)}. \tag{4.8}$$

Proof: Rewrite $g(x)$ as $g(x) = \mathcal{R}\{\sum_{j=1}^{n} a_j e^{i(x+\theta_j)}\} = \mathcal{R}\{e^{ix}(\sum_{j=1}^{n} a_j e^{i\theta_j})\} = \mathcal{R}\{e^{ix} z\}$, where $z \triangleq \sum_j a_j \mathrm{e}^{i\theta_j}$. The minimum occurs when z is rotated to the negative part of the real axis, i.e., $x = \pi - \beta$, and the minimum value is then equal to $-|z|$. This gives expression (4.8). □

Applying Lemma 1 to (4.7) in order to minimize over α, we get

$$\begin{aligned} h(\phi, r) \stackrel{\triangle}{=} \min_{\alpha} f(\alpha, \phi, r) &= c(a + r^2(1-a) + 2rb\cos(\phi)) \\ &+(c-1)[a^2+r^4(1-a)^2+4r^2b^2+2r^2a(1-a)\cos(2\phi) \\ &+4rb(a+r^2(1-a))\cos(\phi)]^{1/2}. \end{aligned}$$

Using the identity $\cos(2\phi) = 2\cos^2(\phi) - 1$ and defining $t \stackrel{\triangle}{=} \cos(\phi)$, we get

$$\begin{aligned} q(t,r) \stackrel{\triangle}{=} h(\cos^{-1}(t), r) &= c(a+r^2(1-a)+2rbt)+(c-1)[a^2+r^4(1-a)^2+4r^2b^2 \\ &-2r^2a(1-a)+4r^2a(1-a)t^2+4rb(a+r^2(1-a))t]^{1/2}. \end{aligned} \tag{4.9}$$

From the definition of t, it follows that $-1/2r \leq t \leq 1/2r$. It can be verified that $q(t, r)$ is a concave function in t. This implies that the minimum of $h(t, r)$ over t is attained at one of the two end points $t = \pm 1/2r$. For these values, and with the variable substitution $\rho = r^2$, we get

$$\begin{aligned} l_{\pm}(\rho) &\stackrel{\triangle}{=} q(\pm 1/2r, r) \\ &= c(a+\rho(1-a)\pm b)+(c-1)[a^2+\rho^2(1-a)^2+4b^2\rho \\ &\quad -2\rho a(1-a)+a(1-a)\pm 2b(a+\rho(1-a))]^{1/2}, \end{aligned} \tag{4.10}$$

where $\rho \geq 1$. $l_+(\rho)$ has "+" instead of $\pm$ and $l_-(\rho)$ has "−". The functions $l_{\pm}(\rho)$ are both concave in ρ. Now, since $l_{\pm}(\rho)$ is the objective function of (4.5), it follows that it must always be positive. Therefore, the minimizer must be $\rho = 1$, which gives that $r = 1$ in (4.9). This implies that the minimum over t in (4.9) occurs at $t = \pm 1/2$, which corresponds to $\phi \in \{\pm\pi/3, \pm 2\pi/3\}$ in

(4.7). This shows that the minimum of $f(\alpha, \phi, r)$ in (4.7) occurs at $r = 1$ and $\phi \in \{\pm\pi/3, \pm 2\pi/3\}$. Inserting these values in the generator matrix $\boldsymbol{B}$, one obtains the generator matrix for the hexagonal lattice as stated in the Theorem. This completes the proof.

While the real-valued case is interesting for theoretical purposes, the complex-valued case is more important for practical MIMO or OFDM applications. Nevertheless, the real-valued result has immediate applications to precoding for mitigation of I/Q imbalance in scalar complex-valued channels.

For the complex-valued case, our main result is:

Theorem 9. *For any non-singular channel matrix $\boldsymbol{S}$, the optimal lattice $\boldsymbol{B}$ in (4.2) is the complex representation of the Schläfli lattice, i.e.,*

$$\boldsymbol{B} = \begin{bmatrix} 1 & \frac{1}{\frac{\pm 1 \pm i}{2}} \\ 0 & \pm\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Proof. It turns out that there is a similar minimum distance preserving condition for complex-valued $\boldsymbol{B}$ as for real-valued ones. In [81], the authors prove that if $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$ and

$$|\mathcal{R}\{\boldsymbol{b}_1^* \boldsymbol{b}_2\}| \leq \frac{1}{2} \text{ and } |\mathcal{I}\{\boldsymbol{b}_1^* \boldsymbol{b}_2\}| \leq \frac{1}{2}, \tag{4.11}$$

then $D_{\min}^2(\boldsymbol{B}) = \|\boldsymbol{b}_1\|^2$. The matrix $\boldsymbol{W}$ is now

$$\boldsymbol{W} = \begin{pmatrix} e^{i(\phi_1 - \gamma_1)} & 0 \\ 0 & e^{i(\phi_3 - \gamma_1)} \end{pmatrix} \begin{pmatrix} \sin(\alpha) e^{-i\phi_1} & \cos(\alpha) e^{-i\phi_2} \\ \cos(\alpha) e^{-i\phi_3} & -\sin(\alpha) e^{-i\phi_4} \end{pmatrix} \tag{4.12}$$

and $\boldsymbol{B}$ is

$$\boldsymbol{B} = \begin{pmatrix} r e^{i\gamma_1} & \sin(\omega) e^{i\gamma_2} \\ 0 & \cos(\omega) e^{i\gamma_3} \end{pmatrix}. \tag{4.13}$$

Hence, $\boldsymbol{W}\boldsymbol{B}$ becomes

$$\boldsymbol{W}\boldsymbol{B} = \begin{pmatrix} r\sin(\alpha) & \sin(\alpha)\sin(\omega) e^{i\theta_1} + \cos(\alpha)\cos(\omega) e^{i\theta_2} \\ r\cos(\alpha) & \cos(\alpha)\sin(\omega) e^{i\theta_1} - \sin(\alpha)\cos(\omega) e^{i\theta_2} \end{pmatrix} \tag{4.14}$$

where $\phi_1 - \phi_2 \equiv \phi_3 - \phi_4 \pmod{2\pi}$, $\theta_1 = \gamma_2 - \gamma_1$ and $\theta_2 = \gamma_3 - \gamma_1 + \phi_1 - \phi_2$. The conditions (4.11) become

$$|\mathcal{R}\{\sin(\omega) e^{-i\theta_1}\}| \leq \frac{1}{2r} \text{ and } |\mathcal{I}\{\sin(\omega) e^{-i\theta_1}\}| \leq \frac{1}{2r}. \tag{4.15}$$

where $r \geq 1$. Define $f(\alpha, \omega, \theta_1, \theta_2, r) \triangleq \mathrm{tr}(\boldsymbol{Z}^* \boldsymbol{B}^* \boldsymbol{W}^* \boldsymbol{S}^{-2} \boldsymbol{W} \boldsymbol{B} \boldsymbol{Z})/s_{2,2}$. We have

$$\begin{aligned}
&f(\alpha,\omega,\theta_1,\theta_2,r) = c[r^2(|z_{11}|^2+|z_{12}|^2)\\
&+|z_{21}|^2+|z_{22}|^2+2\mathcal{R}\{(rz_{11}z_{21}^*+rz_{12}z_{22}^*)\sin(\omega)e^{-i\theta_1}\}]+(1-c)[r^2(|z_{11}|^2+|z_{12}|^2)\\
&-(|z_{21}|^2+|z_{22}|^2)\cos(2\omega)+2\mathcal{R}\{(rz_{11}z_{21}^*+rz_{12}z_{22}^*)\sin(\omega)e^{-i\theta_1}\}]\cos(2\alpha)\\
&-(1-c)[(|z_{21}|^2+|z_{22}|^2)\sin(2\omega)\cos(\theta_1-\theta_2)\\
&+2\mathcal{R}((rz_{11}z_{21}^*+rz_{12}z_{22}^*)\cos(\omega)e^{-i\theta_2})]\sin(2\alpha), \qquad (4.16)
\end{aligned}$$

where $c = (1 + (s_{2,2}/s_{1,1})^2)/2$. First, we minimize over α. It is seen that f depends on α as

$$\begin{aligned}
f(\alpha,\omega,\theta_1,\theta_2,r) &= a_1 + a_2\cos(2\alpha) + a_3\sin(2\alpha)\\
&= a_1 + \sqrt{a_2^2+a_3^2}\left(\frac{a_2}{\sqrt{a_2^2+a_3^2}}\cos(2\alpha) + \frac{a_3}{\sqrt{a_2^3+a_3^2}}\sin(2\alpha)\right)\\
&= a_1 + \sqrt{a_2^2+a_3^2}(\sin(\psi)\cos(2\alpha)+\cos(\psi)\sin(2\alpha))\\
&= a_1 + \sqrt{a_2^2+a_3^2}\sin(2\alpha+\psi), \qquad (4.17)
\end{aligned}$$

where the constants a_1, a_2 and a_3 are easily read of from (4.16) and ψ is such that $\sin(\psi) = a_2/\sqrt{a_2^2+a_3^2}$. The minimum of (4.17) over α occurs at $\alpha = -\pi/4 - \psi/2$, which gives $f(-\pi/4-\psi/2, \omega, \theta_1, \theta_2, r) = a_1 - \sqrt{a_2^2+a_3^2}$. Since only a_3 depends on θ_2, minimizing f over θ_2 implies maximizing a_3^2 over θ_2. We have

$$\begin{aligned}
a_3 &= -(1-c)[(|z_{21}|^2+|z_{22}|^2)\sin(2\omega)\cos(\theta_1-\theta_2)\\
&\quad +2\mathcal{R}((rz_{11}z_{21}^*+rz_{12}z_{22}^*)\cos(\omega)e^{-i\theta_2})]\\
&= -(1-c)\mathcal{R}\{e^{-i\theta_2}((|z_{21}|^2+|z_{22}|^2)\sin(2\omega)e^{i\theta_1}+2\cos(\omega)(rz_{11}z_{21}^*+rz_{12}z_{22}^*))\}.
\end{aligned}$$

It follows that the maximizing θ_2 is such that $e^{i\theta_2}$ rotates the expression it multiplies to the real axis. We get

$$\begin{aligned}
&\min_{\theta_2} f(-\pi/4-\psi/2,\theta_1,\theta_2,\omega,r) = l(\theta_1,\omega,r) =\\
&c[r^2(|z_{11}|^2+|z_{12}|^2)+|z_{21}|^2+|z_{22}|^2 + 2\mathcal{R}\{\sin(\omega)e^{-i\theta_1}(rz_{11}z_{21}^*+rz_{12}z_{22}^*)\}]\\
&+(c-1)[\big(r^2(|z_{11}|^2+|z_{12}|^2)+|z_{21}|^2+|z_{22}|^2\\
&+ 2\mathcal{R}\{\sin(\omega)e^{-i\theta_1}(rz_{11}z_{21}^*+rz_{12}z_{22}^*)\}\big)^2 - 4\cos^2(\omega)|\det(\boldsymbol{Z})|^2]^{1/2}. \qquad (4.18)
\end{aligned}$$

As in the real-valued case, it can easily be shown that the expression in (4.18) is concave in $\sin(\omega)$. Thus, the minimum is attained at the endpoints of

$\sin(\omega)$. The constraints in (4.15) can be written as $|\sin(\omega)\cos(\theta_1)| \leq 1/2r$ and $|\sin(\omega)\sin(\theta_1)| \leq 1/2r$. Assume $|\sin(\theta_1)| \leq |\cos(\theta_1)|$. It follows that the interval for $\sin(\omega)$ is $-1/(2r\cos(\theta_1)) \leq \sin(\omega) \leq 1/(2r\cos(\theta_1))$, while the interval for θ_1 is $-\pi/4 \leq \theta_1 \leq \pi/4$. Inserting either one of these endpoints for $\sin(\omega)$ in (4.18) and using the trigonometric identity $1/\cos^2(x) = 1 + \tan^2(x)$, we get that l takes on the following form

$$\begin{aligned} l(\theta_1, r) &= c(b_1 + b_2\tan(\theta_1)) \\ &\quad + (c-1)\left[(b_1 + b_2\tan(\theta_1))^2 + \frac{|\det(\boldsymbol{Z})|^2}{r^2}\tan^2(\theta_1)\right. \\ &\quad \left. + |\det(\boldsymbol{Z})|^2(4 - 1/r^2)\right], \end{aligned} \tag{4.19}$$

where b_1 and b_2 are constants with respect to θ_1. Again, it is clear that (4.19) is concave in $\tan(\theta_1)$, and thus the minimum is attained at one of the endpoints of θ_1, which are $-\pi/4$ and $\pi/4$. If we instead assumed that $|\sin(\theta_1)| \geq |\cos(\theta_1)|$, the only difference is that $\tan(\theta_1)$ becomes $\cot(\theta_1)$ and $\pi/4 \leq \theta_1 \leq 3\pi/4$. This gives rise to the same behavior of $l(\theta_1, r)$ and thus same results are obtained.

To recap, we showed that the minimum for $l(\theta_1, \omega, r)$ in (4.18) over θ_1, ω occurs when $\theta_1 = \pm\pi/4$ and at the endpoints for $\sin(\omega)$, which are then $\sin(\omega) = \pm 1/(2r\cos(\theta_1)) = \pm 1/\sqrt{2}r$. We now continue by inserting this expression for $\sin(\omega)e^{-i\theta_1}$ in (4.18) and obtain a one-dimensional function in $\rho = r^2$ of the form

$$l_1(\rho) = k_1 + k_2\rho + (c-1)\sqrt{k_3\rho^2 + k_4\rho + k_5 + |\det(\boldsymbol{Z})|^2(2/\rho - 4)}, \tag{4.20}$$

where the k_j are constants with regard to ρ and with k_3 positive. If we instead study the function $l_2(\rho) = k_1 + k_2\rho + (c-1)\sqrt{k_3\rho^2 + k_4\rho + k_5 - 2|\det(\boldsymbol{Z})|^2}$, it follows from the same concavity arguments as before that $l_2(\rho)$ is a concave function and thus the minimum is attained at the endpoints, which are $\rho = 1$ and $\rho = \infty$. From the concavity of $l_2(\rho)$ it follows that if the minimum is attained at ∞, then the minimum value is $-\infty$, which is impossible since the trace function is always positive; thus the minimum of $l_2(\rho)$ must be attained at $\rho = 1$. Now comparing $l_2(\rho)$ with $l_(\rho)$, the only difference is the term $|\det(\boldsymbol{Z})|^2(2/\rho - 4)$ in the square root, with maximum value of $2|\det(\boldsymbol{Z})|^2$ attained at $\rho = 1$; hence $l_2(1) = l_1(1)$. Since $c-1$ is always non-positive, it follows that $l_2(\rho) \leq l_1(\rho)$ for $\rho \geq 1$, which gives that the minimum of $l_1(\rho)$ occurs when $\rho = r = 1$ (because the minimum of $l_2(\rho)$ occurs for $\rho = 1$).

We have now showed that the minimum of $l(\theta_1, \omega, r)$ in (4.18) occurs for $\theta_1 = \pm\pi/4$, $\sin(\omega) = \pm 1/\sqrt{2}r$, $r = 1$. Inserting these values into the lattice

generator $\boldsymbol{B}$ in (4.13), we arrive to the following optimal lattice generator

$$\boldsymbol{B} = \begin{pmatrix} 1 & \frac{\pm 1 \pm i}{2} \\ 0 & \pm\frac{1}{\sqrt{2}} \end{pmatrix}. \tag{4.21}$$

Extending $\boldsymbol{B}$ to its real-valued representation by means of (3.22), it holds that for each realization of $\pm$ as $+$ or $-$, that $\boldsymbol{B}_r$ is a generator matrix for the Schläfli lattice D4. □

Hence, by "complex representation", it is meant that if the transformation (3.22) is performed on $\boldsymbol{B}$ in Theorem 9, the Schläfli lattice in four real-valued dimensions results. Its real-valued generator matrix is

$$D_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \tag{4.22}$$

To summarize, the two dimensional minimum distance optimal precoder for "large" input constellations is always an instance of the hexagonal or the Schläfli lattice for real-valued and complex-valued precoding, respectively.

4.2.2 Optimal $\boldsymbol{Z}$ matrix

Since $\boldsymbol{B}$ is now known, it remains to find the optimal basis matrix $\boldsymbol{Z}$ in order to solve (4.2). This section describes the core idea of the algorithms that find the optimal real-valued and complex-valued $\boldsymbol{Z}$, respectively. A complete Matlab code for the algorithms can be found at *www.eit.lth.se/goto/Zalgorithm.*

By inserting the optimal real-valued $\boldsymbol{B}$ and $\boldsymbol{W}$ into (4.2), the optimization (4.2) is equivalent to[1]

$$\boldsymbol{Z} = \arg\min_{\boldsymbol{Z}} \mu_{\pm}^{r}(\boldsymbol{Z}),$$

where

$$\begin{aligned} \mu_{\pm}^{r}(\boldsymbol{Z}) &\stackrel{\triangle}{=} s_{2,2}(z_{11}^2+z_{12}^2+z_{21}^2+z_{22}^2)l_{\pm}(1) \\ &= c[z_{11}^2+z_{12}^2+z_{21}^2+z_{22}^2 \pm (z_{11}z_{21}+z_{12}z_{22})] \\ &\quad +(c-1)[(z_{11}^2+z_{12}^2)^2+(z_{21}^2+z_{22}^2)^2+4(z_{11}z_{21}+z_{12}z_{22})^2 \\ &\quad -(z_{11}^2+z_{12}^2)(z_{21}^2+z_{22}^2) \pm 2(z_{11}z_{21}+z_{12}z_{22})(z_{11}^2+z_{12}^2+z_{21}^2+z_{22}^2)]^{1/2}. \end{aligned} \tag{4.23}$$

[1]The optimization over $\boldsymbol{W}$ is treated in the proofs of Theorem 8 and 9.

In the complex-valued case, we have the following optimization

$$\boldsymbol{Z} = \arg\min_{\boldsymbol{Z}} \mu_{\pm}^{c}(\boldsymbol{Z}),$$

where

$$\begin{aligned}\mu_{\pm}^{c}(\boldsymbol{Z}) \triangleq\ & c(\|\boldsymbol{Z}\|^2+\mathcal{R}\{(\pm 1 \pm i)(z_{11}z_{21}^*+z_{12}z_{22}^*)\}) \\ & + (c-1)\sqrt{(\|\boldsymbol{Z}\|^2+\mathcal{R}\{(\pm 1 \pm i)(z_{11}z_{21}^*+z_{12}z_{22}^*)\})^2-2}.\end{aligned} \tag{4.24}$$

The $\pm$ signs in both (4.23) and (4.24) can be absorbed into the elements of $\boldsymbol{Z}$, without changing the unimodularity of $\boldsymbol{Z}$. Define $\beta_r \triangleq z_{11}^2+z_{12}^2+z_{21}^2+z_{22}^2-(z_{11}z_{21}+z_{12}z_{22})$ and $\beta_c \triangleq |z_{11}|^2+|z_{12}|^2+|z_{21}|^2+|z_{22}|^2+\mathcal{R}\{(1+i)(z_{11}z_{21}^*+z_{12}z_{22}^*)\}$, where we do not expliticly denote the dependency of β_r and β_c on $\boldsymbol{Z}$. Since $|\det(\boldsymbol{Z})| = 1$, (4.23) and (4.24) become

$$\mu^r(\beta_r) = c\beta_r + (c-1)\sqrt{\beta_r^2 - 3} \tag{4.25}$$

and

$$\mu^c(\beta_c) = c\beta_c + (c-1)\sqrt{\beta_c^2 - 2}, \tag{4.26}$$

respectively. The difference between (4.25) and (4.23) (similarly between (4.26) and (4.25)) is that the former only depends on one variable, that implicitly depends on the elements $\{z_{ij}\}$, while the latter is directly expressed in the elements $\{z_{ij}\}$. Deriving the optimal β_r and β_c does not produce the optimal elements $\{z_{ij}\}$, however, it can provide easier optimality conditions for $\{z_{ij}\}$. If we for the moment drop the constraint that β_r has to be integer-valued, the function $\mu^r(\boldsymbol{Z})$ in (4.25) will be minimized over β_r. It can be verified that $\mu^r(\boldsymbol{Z})$ is a convex function. Differentiating $\mu(\beta_r)$ with respect to β and setting the derivative to 0 gives that $\beta_{r,\text{opt}} = \sqrt{\frac{3c^2}{2c-1}}$ is the optimal point. Since $\mu^r(\boldsymbol{Z})$ is convex, the minimum of $\mu(\beta_r)$ over unimodular matrices can only occur at two specific matrices. Either it is the $\boldsymbol{Z}$ that produces the largest β_r smaller than $\beta_{r,\text{opt}}$, or it is the $\boldsymbol{Z}$ that produces the smallest β_r larger than $\beta_{r,\text{opt}}$. A similar analysis can be applied to the complex-valued (4.26), and it follows that the largest β_c smaller, or smallest β_c larger, than $\beta_{c,\text{opt}} = \sqrt{2}c/\sqrt{2c-1}$ is optimal. Hence, in the real-valued case, an algorithm can be developed that traverses unimodular $\boldsymbol{Z}$'s and stops when two matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are found, such that $\boldsymbol{Z}_1$ gives the β_r that equals the largest integer smaller than $\beta_{r,\text{opt}}$, and $\boldsymbol{Z}_2$ gives the β_r that equals the smallest integer larger than $\beta_{r,\text{opt}}$. An algorithm for the complex-valued case works in the same way. Due to lack of space and the fact that the algorithms are ad-hoc, we omit the implementation

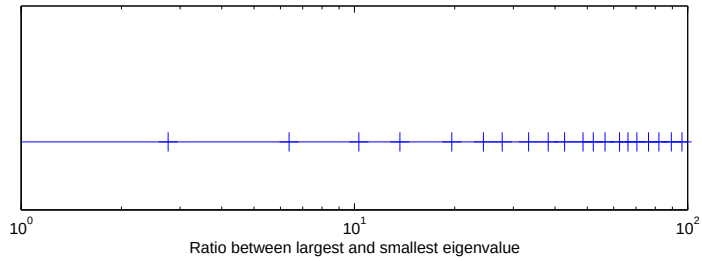

Figure 4.3: Change in $\boldsymbol{Z}$ with respect to the ratio $s_{1,1}/s_{2,2}$. The solution to (4.2) is constant for all $\boldsymbol{S}$ with a ratio between any two consecutive markers. The scale on the x-axis is logarithmic.

details and refer to *www.eit.lth.se/goto/Zalgorithm* where the Matlab code for both algorithms can be found.

Since we now know that solving (4.2) is a discrete optimization problem, it is of interest to see how often the solution changes with varying $\boldsymbol{S}$. Figure 4.3 shows the ratio $s_{1,1}/s_{2,2}$ on the x-axis, and the markers show the ratios where $\boldsymbol{Z}$ changes. As seen, the same solution can be used for a wide interval.

4.2.3 Applications

In this section we consider a number of practical applications of the optimal minimum distance lattice based precoder and make comparisons to other schemes. As discussed in Section 2.5.2, minimum distance based precoders are asymptotically optimal in the high SNR regime, but minimum distance plays little role at low SNR, so significant performance gains cannot be expected there.

Consider first the 2×2 channel studied in [83],

$$\boldsymbol{S} = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.27}$$

In [83], this channel was studied at asymptotically high SNR for binary baseband alphabets with real-valued precoding. The objective was to find the real-valued precoder $\boldsymbol{F}$ that maximizes the mutual information $I(\boldsymbol{SFx}+\boldsymbol{n};\boldsymbol{x})$. For high SNR, it is known that the optimal mutual information precoder converges to the optimal minimum distance precoder, and the numerical optimization framework in [83] thus produced the optimal minumum distance precoder. The

precoder is of the following simple form

$$\boldsymbol{F} = \begin{bmatrix} \sqrt{2} & \sqrt{2} \\ -\sqrt{2} & \sqrt{2} \end{bmatrix}. \tag{4.28}$$

It can be verified by standard techniques that the combined channel-precoder matrix $\boldsymbol{SF}$ is an instance of the hexagonal lattice - which is precisely the result if an infinite lattice constellation was used. For such a lattice constellation, the strength of the results in Theorem 8 and 9 is that no numerical optimization of the precoder is necessary since it is known a-priori that the hexagonal lattice *must* be the solution, and it only remains to find the optimal basis matrix $\boldsymbol{Z}$ according to the algorithm mentioned in Section 4.2.2. By doing so, we find that the optimal $\boldsymbol{Z}$ for asymptotically large constellations coincides with the basis matrix that is built into (4.28). Altogether, for the particular channel (4.27) studied in [83], a "large" constellation means binary and it is known beforehand what structure the solution must have.

In Figure 4.4 we continue to study the channel (4.27), but now by evaluating its mutual information that is achieved by 4QAM inputs when the complex-valued minimum distance optimal precoder for large constellations is used. As comparisons, plots of the achieved mutual information for 1) no precoding at all, i.e., $\boldsymbol{F}=\boldsymbol{I}$, 2) Mercury/Waterfilling from [82], and 3) capacity achieved by Gaussian inputs and waterfilling, are presented. The performance of the optimal mutual information precoder coincides with that of Mercury/Waterfilling in the low SNR regime, while it coincides with that of the minimum distance precoder in the high SNR regime. As can be seen, there is a 2 dB gain offered by the minumum distance precoder over uncoded systems and Mercury/Waterfilling at high SNR. At low SNR, the Mercury/Waterfilling policy is optimal and outperforms the minimum distance precoder.

For the channel (4.27), we observed that the large constellation assumption made in this chapter was not very critical as it produced the same result as a binary input constellation does. This is, however, not true in general, and it is necessary to investigate the impact of the cardinality of the input constellation. Consider diagonal channel matrices $\boldsymbol{S}$ where each diagonal element is a zero-mean, unit-variance, circulary symmetric complex Gaussian random variable ($\mathcal{CN}(0,1)$). The average mutual information, against SNR, is computed for 4QAM and 16QAM input constellations for 1) the minimum distance optimal precoder for large constellations, 2) minimum distance optimal precoders for the particular constellations used, and 3) no precoder. The average is evaluated over 10^6 channel realizations by straightforward Monte Carlo simulation. For 4QAM and 16QAM, the minimum distance optimal precoders have been reported in [67, 72], while the optimal precoder for 64QAM has so far not been reported in the literature which is the reason why we do not go beyond 16QAM.

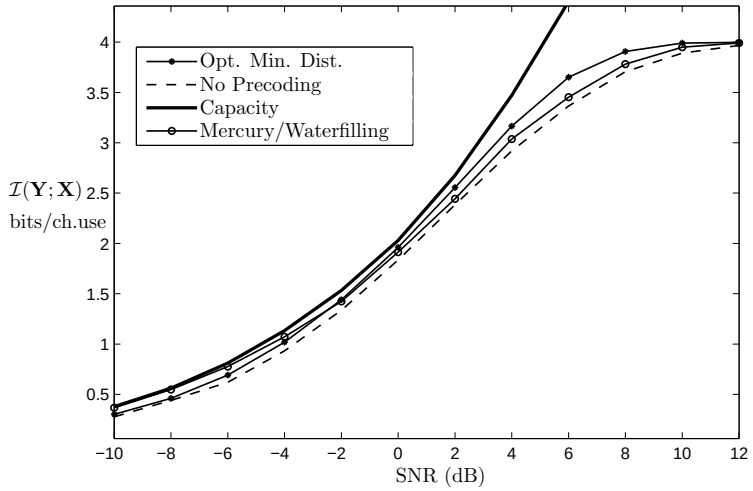

Figure 4.4: Mutual information for the channel (4.27) studied in [83] with 4QAM inputs under different settings. The solid heavy line shows the capacity with waterfilling, the curve marked with asterixes shows the ensuing mutual information from the lattice precoder in this section and the curve marked with circles show the Mercury/Waterfilling mutual information. The bottom line is the no precoding case.

The results are shown in Figure 4.5. The uppermost heavy solid line corresponds to the average capacity of the channel achieved by Gaussian inputs with waterfilling. The lower set of curves corresponds to 4QAM while the upper corresponds to 16QAM. Within each set of curves, the lower curve (without markers) shows the no precoder case, the middle curve (marked with asterixes) is the performance of the precoder constructed from a large constellation assumptions, and the upper curve (marked with circles) is the performance of the precoder explicitly constructed for the input constellation used. For 4QAM inputs, a small loss of the large constellation construction can be seen, while for 16QAM the ensuing mutual information from a large constellation assumption is virtually indistinguishable from that of a construction expliticitly made for 16QAM. Hence, it can be concluded from this example that a 16QAM input constellation can be replaced by an infinite lattice constellation without appreciably affecting the results. This greatly simplifies the precoder optimization problem since lattice theoretic tools can be applied.

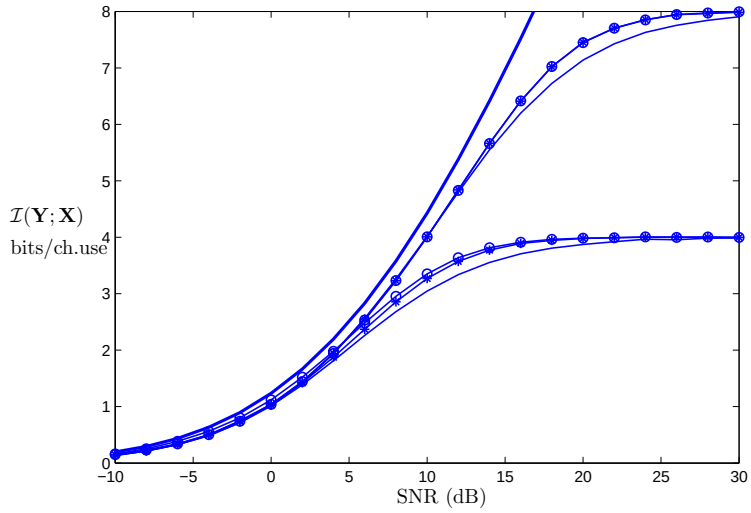

Figure 4.5: Average mutual information for random diagonal channels with 4QAM (bottom set) and 16QAM (upper set). The heavy solid line is the capacity with waterfilling. Within each set, the line marked with circles shows the performance of a precoder constructed expliticly for the input constellation used, and the curve marked with asterixes shows the performance of the precoder constructed from an infinite lattice constellation assumption. These two curves are virtually identical for 16QAM. The bottom line within each set corresponds to the no precoding case.

In Figure 4.6 we turn our attention towards the error probability of 2×2 MIMO systems with 1) the minimum distance optimal precoder for large constellations, 2) minimum distance optimal precoders for the particular constellations used, and 3) no precoding. 4QAM, 16QAM, and 64QAM input constellations, together with a maximum likelihood detector, are considered. The lines marked with circles correspond to the minimum distance optimal precoder for large constellations, the lines marked with squares correpond to the optimal precoder designed for the particular input constellations used, and the unmarked lines correspond to the no-precoder case. As can be seen, there is a large gain from explicitly taking the input constellation into account for 4QAM. However, for 16QAM inputs, this gain reduces significantly, so that the precoder designed for large constellations performs close to optimal. For 64QAM, the gap to the optimal precoder designed expliticly for 64QAM can

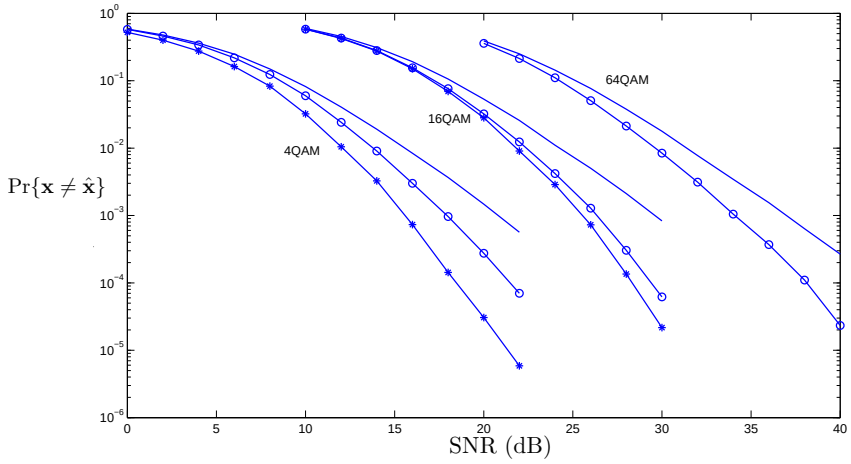

Figure 4.6: Maximum likelihood receiver tests of various precoders with 4QAM, 16 QAM, and 64QAM. Within each set, the rightmost curve is the no precoding case, the middle curve is the precoder constructed from an infinite lattice constellation assumption, and the leftmost curve is the performance of a precoder constructed explitiely for the input constellation used (not present for 64QAM).

not be determined. However, given the large reduction of the gap between the 4QAM and 16QAM cases, we expect that the gap for 64QAM is minor, so that the precoder designed for large constellations is virtually optimal.

As a final example, an OFDM system with N sub-carriers $\{H_k\}_{k=1}^N$ is investigated. For simplicity, all sub-carriers are assumed to be independent zero-mean, unit-variance, circulary symmetric complex Gaussian random variables ($\mathcal{CN}(0,1)$). In practice, adjacent carriers are strongly correlated but for the transceiver system to be considered, N is large and such correlations are immaterial. The approach taken in [65] is pursued, but now with the 2×2 minimum distance optimal precoder constructed from the large constellation assumption as a building block to construct much larger precoder structures. The N sub-carriers are first grouped into $N/2$ pairs. The particular pairing used in [65] is to combine the strongest sub-carrier with the weakest sub-carrier, the second strongest with the second weakest etc. Let $\{\tilde{H}_k\}_{k=1}^N$ denote the sub-carriers $\{H_k\}_{k=1}^N$, but sorted according to their strengths so that

$|\tilde{H}_1| \geq |\tilde{H}_1| \geq \ldots \geq |\tilde{H}_N|$. We have $N/2$ independent transmissions

$$\boldsymbol{y}_k = \begin{bmatrix} \tilde{H}_k & 0 \\ 0 & \tilde{H}_{N-k+1} \end{bmatrix} \boldsymbol{F}_k + \boldsymbol{n}_k = \tilde{\boldsymbol{S}}_k \boldsymbol{F}_k + \boldsymbol{n}_k, \qquad 1 \leq k \leq N/2$$

and we need to construct $N/2$ precoders $\{\boldsymbol{F}_k\}_{k=1}^{N/2}$. A total energy of $N\,P/2$ is assumed, and we allocate a fraction γ_k to $\boldsymbol{F}_k$ under the constraint that $\sum \gamma_k = N\,P/2$. Our power allocation policy is that all channel-precoder pairs $\tilde{\boldsymbol{S}}_k \boldsymbol{F}_k$ should have equal minimum distances. We can find the precoders according to this policy as follows:

- Design $\{\boldsymbol{F}_k\}_{k=1}^{N/2}$ according to the constraint $\mathrm{Tr}(\boldsymbol{F}_k^* \boldsymbol{F}_k) = 1$.
- From lattice theory, it is guaranteed that the minumum distance for each channel-precoder pair equals the length of the shortest vector of the lattice spanned by $\tilde{\boldsymbol{S}}_k \boldsymbol{F}_k$. Let D_k^2 denote the minimum distance.
- The power allocation that equalizes all minumum distance is proportional to
$$\gamma_k \propto \frac{1}{D_k^2}$$
and the overall power constraint $\sum \gamma_k = N\,P/2$ finally yields the set of precoders.

The ensuing average mutual information of this strategy is compared with the no-precoder case, Mercury/Waterfilling, and the capacity of the channel. The input constellation is 16QAM in all cases (except for the capacity case where it is complex Gaussian). The results are shown in Figure 4.7. Note that the average mutual information per channel-precoder pair is plotted. The top heavy solid curve is the average capacity of the channel, the curve marked by circles is the system based on the minimum distance optimal precoder described above, the curve marked with asterixes is the Mercury/Waterfilling system, and the bottom curve shows the performance of the no-precoder case. As in the previous examples, there are no gains at low-moderate SNR by the minumum distance optimal precoder, while the gains are significant at high SNR. Note that the Mercury/Waterfilling is close to optimal at low SNR while it suffers from large penalties at high SNR.

4.3 Optimal Lattice Precoders for Arbitrary Dimensions

This section will extend the real-valued results in Section 4.2 to arbitrary dimensions. First, (3.5) is transformed into an equivalent real-valued model, by

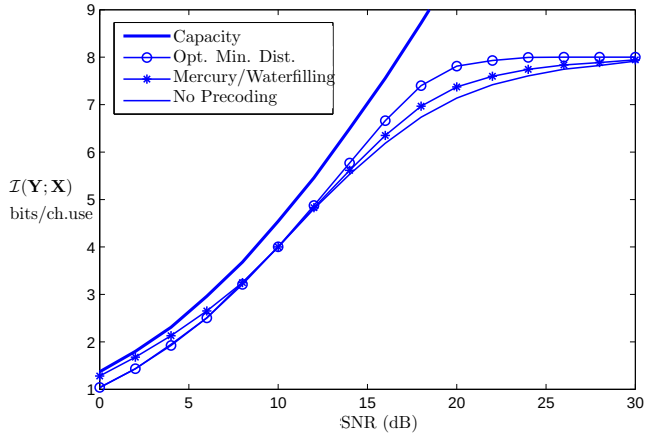

Figure 4.7: Average mutual information per sub-carrier pair with 16QAM inputs under different settings. The solid heavy line shows the capacity with waterfilling, the curve marked with circles shows the ensuing mutual information from the lattice precoder in this section and the curve marked with asterixes shows the mercury/waterfilling mutual information. The bottom line is the no precoding case.

means of the transformations (3.22) and (3.23). Thus, there is no loss of generality by assuming real-valued systems, since they can represent complex-valued systems as well. However, it will soon be evident that the real-valued representation has an inherent gain. Applying the transformations in (3.22) and (3.23) to the matrices and vectors in (3.5) yields the real-valued model

$$\boldsymbol{y}_r = \boldsymbol{S}_r \underbrace{\boldsymbol{F}_r \boldsymbol{a}_r}_{\boldsymbol{x}_r} + \boldsymbol{n}_r. \tag{4.29}$$

Since $\boldsymbol{a}$ can be any Gaussian integer vector of dimension N, the real-valued vector $\boldsymbol{a}_r$ can be any integer vector of dimension $2N$. Further, it holds that $\mathrm{tr}(\boldsymbol{F}_r^{\mathrm{T}} \boldsymbol{F}_r) = 2\mathrm{tr}(\boldsymbol{F}^* \boldsymbol{F}) \leq 2P_0$. The precoding is now performed over the real-valued domain as $\boldsymbol{x}_r = \boldsymbol{F}_r \boldsymbol{a}_r$, where the actual complex-valued symbols $\boldsymbol{x}$ to be transmitted over $\boldsymbol{S}$ in (3.5) are obtained from $\boldsymbol{x}_r$ through the inverse of (3.23). Note that the transformation in (3.22) imposes a skew-symmetric structure on $\boldsymbol{F}_r$, which can be relaxed when the precoding is performed in the real-valued domain, i.e., $\boldsymbol{F}_r$ can be any $2N \times 2N$ real-valued matrix satisfying the trace constraint. Thus, by precoding over the real-valued domain, performance

gains can be expected because there are more degrees of freedom in designing $\boldsymbol{F}_r$ than in designing $\boldsymbol{F}$. Henceforth, we omit the subscript r and assume that all variables in N dimensions are real-valued, unless stated otherwise.

For completeness, the problem in (3.6) is restated again, but now in real-valued terms:

$$\begin{aligned} &\min_{\boldsymbol{F}} \operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^T) \\ &\text{subject to} \\ &\boldsymbol{e}^T\boldsymbol{G}\boldsymbol{e} \geq 1 \quad \forall \boldsymbol{e} \in \mathbb{Z}^N \backslash \{\boldsymbol{0}_N\}, \end{aligned} \tag{4.30}$$

where $\mathbb{Z}^N \backslash \{\boldsymbol{0}_N\}$ is the set of all N-dimensional integer vectors except the all-zero vector. Yet another equivalent way of expressing (4.30) is to maximize the *normalized minimum distance* $d^2_{\min}(\boldsymbol{S}\boldsymbol{F}) \stackrel{\triangle}{=} D^2_{\min}(\boldsymbol{S}\boldsymbol{F})/\operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^{\mathrm{T}})$ over $\boldsymbol{F} \neq \boldsymbol{0}_{N\times N}$. For our purposes, the problem formulation in (4.30) will turn out to be the most convenient, and will be focus of study. In Section 4.3.1, we formulate (4.30) as a pure lattice problem, and introduce the tools from lattice theory needed to analyze it.

4.3.1 Lattice-theoretic approach

This section is split into four parts. Section 4.3.1 describes the Ryshkov polytope and Section 4.3.1 the Minkowski polytope, both of fundamental importance for the understanding of the subsequent analysis. Section 4.3.2 formulates (4.30) as a lattice problem, while Section 4.3.3 gives an overview of famous lattice problems and techniques, applicable to the minimum distance problem, to solve them.

Some terms from convex geometry will be used in what follows. The set $\{\lambda_1\boldsymbol{v}_1 + \ldots \lambda_k\boldsymbol{v}_K \ : \ \lambda_j \geq 0,\, 1 \leq j \leq K,\}$, for K given N-dimensional points $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K$, is called an N-dimensional *polyhedral cone*. A *polytope* in N dimensions is the intersection of a finite number of N-dimensional halfspaces[2], i.e., the set of N-dimensional points $\{\boldsymbol{x} \ : \ a_{j,1}x_1 + \ldots a_{j,N}x_N \leq b_j \ : \ 1 \leq j \leq M\}$ for given numbers M, $a_{j,i}$, b_j, $1 \leq i \leq N$, $1 \leq j \leq M$. A *face* of a polytope is the intersection between the polytope and a supporting hyperplane[3] of the polytope. If the face is one-dimensional, we call it an *edge*, or in the case when the polytope is a polyhedral cone, an *extreme ray*.

[2]By a half-space we mean either of the two parts into which a hyperplane divides a Euclidean space.

[3]A supporting hyperplane of a set $\mathcal{S}$ is a hyperplane that intersects $\mathcal{S}$, such that $\mathcal{S}$ is completely contained in one of the two halfspaces determined by the hyperplane.

Ryshkov polytope

Let $\Lambda_{\boldsymbol{L}} \subset \mathbb{R}^N$ be a lattice with a generator matrix $\boldsymbol{L}$ and with $D^2_{\min}(\boldsymbol{L}) \geq \lambda$. This can be written as an infinite set of inequalities $\boldsymbol{e}^T \boldsymbol{G}_{\boldsymbol{L}} \boldsymbol{e} \geq \lambda$, where $\boldsymbol{e} \in \mathbb{Z}^N / \{\boldsymbol{0}_N\}$ and $\boldsymbol{G}_{\boldsymbol{L}} = \boldsymbol{L}^{\mathrm{T}} \boldsymbol{L}$. Since $\boldsymbol{G}_{\boldsymbol{L}}$ is a symmetric matrix, its dimension is $N(N+1)/2$, and the infinite set of inequalities are linear over the $N(N+1)/2$ distinct elements of $\boldsymbol{G}$. By considering the distinct elements in $\boldsymbol{G}$ as a vector $(g_{1,1}, \ldots, g_{1,N}, g_{2,2}, \ldots, g_{2,N}, \ldots, g_{N,N})$ in $\mathbb{R}^{N(N+1)/2}$, the infinite set of inequalities represent an intersection of infinitely many halfspaces in $\mathbb{R}^{N(N+1)/2}$.

Definition 7. *The* Ryshkov polytope $\mathcal{R}_\lambda$ *is the set* $\mathcal{R}_\lambda \stackrel{\triangle}{=} \{\boldsymbol{G} : \boldsymbol{e}^{\mathrm{T}} \boldsymbol{G} \boldsymbol{e} \geq \lambda,\ \boldsymbol{e} \in \mathbb{Z}^N / \{\boldsymbol{0}_N\}\}$.

It is easily realized that any $\boldsymbol{G} \in \mathcal{R}_\lambda$ is positive definite, thus $\mathcal{R}_\lambda \subset \mathcal{S}^{N \times N}_{\succ 0}$. In the vector space $\mathbb{R}^{N(N+1)/2}$, the set of positive definite matrices $\mathcal{S}^{N \times N}_{\succ 0}$ corresponds to a cone, where $\mathcal{R}_\lambda$ is contained in the interior of the cone. $\mathcal{R}_\lambda$ is a convex and unbounded set, since if $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathcal{R}_\lambda$, then $k_1 \boldsymbol{G}_1 + k_2 \boldsymbol{G}_2 \in \mathcal{R}_\lambda$ for $k_1, k_2 \geq 0$ and $k_1 + k_2 \geq 1$. Because any positive definite Gram matrix $\boldsymbol{G}$, hereinafter called a "positive quadratic form" (PQF), corresponds to a lattice, the Ryshkov polytope contains all Gram matrices of lattices with minimum distance of at least λ.

Since $\mathcal{R}_\lambda$ is the intersection of infinitely many halfspaces, it could be the case that $\mathcal{R}_\lambda$ has a boundary that is "curved" and does not represent a polytope. More formally, there could exist a point on the boundary of $\mathcal{R}_\lambda$ for which there is only one support plane, which intersects $\mathcal{R}_\lambda$ only at this point. We say that an intersection of infinitely many halfspaces $\mathcal{P} = \cap_{i=1}^{\infty} \mathcal{H}_i$, is a *locally finite polytope*, if the intersection of $\mathcal{P}$ and an arbitrary polytope is again a polytope. Thus, a locally finite polytope $\mathcal{P}$ contains no curved boundary. The following theorem [84] justifies the name "Ryshkov polytope", and plays a fundamental role for the classification of optimal precoders that is developed in this work.

Theorem 10. *For $\lambda > 0$, the set $\mathcal{R}_\lambda$ is a locally finite polytope.*

A vertex in $\mathcal{R}_\lambda$ corresponds to a form $\boldsymbol{G}$ that is the unique solution to a set of at least $N(N+1)/2$ linearly independent equations $\boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{G} \boldsymbol{e}_j = \lambda$, $j = 1 \ldots K$, where $K \geq N(N+1)/2$. Note that if $\boldsymbol{G}$ is a vertex in $\mathcal{R}_\lambda$, then so is $\boldsymbol{Z}^{\mathrm{T}} \boldsymbol{G} \boldsymbol{Z}$ where $\boldsymbol{Z}$ is unimodular, and thus there is an infinite, but countable, number of vertices in the Ryshkov polytope. This observation also implies that the vertices can be partitioned into equivalence classes, where the equivalence relation is an isometry between two vertices. A conceptual visualization of $\mathcal{R}_\lambda$ is given in Figure 4.8.

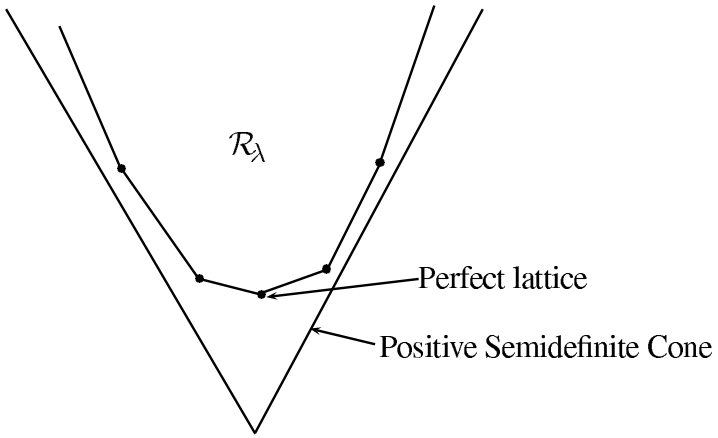

Figure 4.8: A conceptual visualization of the Ryshkov polytope $\mathcal{R}_\lambda$. The dots correspond to vertices in $\mathcal{R}_\lambda$, i.e., the perfect forms. The boundary of the positive semidefinite cone never intersects with $\mathcal{R}_\lambda$, since this boundary consists of forms with arbitrarily small minimum distance. Moreover, this boundary is not made up of straight lines as the simplified figure shows in two dimensions, but is a complicated surface in higher dimensions.

Lattices corresponding to vertices of $\mathcal{R}_\lambda$ are named *perfect lattices* in the literature [85], and the corresponding Gram matrices are *perfect forms*. The next theorem gives another interesting property of the Ryshkov polytope [85], which is important for this work.

Theorem 11. *There are only finitely many non-isometric perfect forms in the Ryshkov polytope.*

Hence, although there are infinitely many perfect forms in the Ryshkov polytope, Theorem 11 reveals that out of these, only finitely many are non-isometric and correspond to different lattices. The non-isometric perfect lattices have been tabulated for all dimensions up to $N = 8$ [84]. In two and three dimensions, there is only one unique perfect lattice. In four dimensions, there are two, in five there are three, and in 8 dimensions there are 10916.

Voronoi's algorithm [85], is commonly used to traverse the vertices of the Ryshkov polytope. In Section 4.3.6 we present an algorithm for solving our problem, which in essence is a modified version of Voronoi's algoritm.

Minkowski polytope

Another characterization of PQFs is via *Minkowski reduction*.

Definition 8. *The Minkowski reduction region $\mathcal{M}$ is the set of all $\boldsymbol{G}$ satisfying*

$$\begin{aligned} &(i) \quad \boldsymbol{v}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{v} \geq g_{i,i}, \quad \text{for all } \boldsymbol{v} \in \mathbb{Z}^N \text{ such that } \gcd(v_i, \ldots, v_N) = 1. \\ &(ii) \quad g_{i,i+1} \geq 0, \quad i = 1, \ldots, N-1. \end{aligned} \tag{4.31}$$

As with the Ryshkov polytope, $\mathcal{M}$ is a subset of $\mathcal{S}_{\succ 0}^{N\times N}$. A PQF $\boldsymbol{G} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L} \in \mathcal{M}$ is said to be *Minkowski reduced* and the lattice generator matrix $\boldsymbol{L}$ is called a *Minkowski reduced* generator matrix for $\Lambda_{\boldsymbol{L}}$. It can be shown that any lattice $\Lambda_{\boldsymbol{B}}$ has a generator matrix $\boldsymbol{L}$ that is Minkowski reduced, i.e., there exists an $\boldsymbol{L}$ such that $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{Z}$, where $\boldsymbol{L}$ is Minkowski reduced and $\boldsymbol{Z}$ is a unimodular matrix [86]. Note that the Minkowski reduced generator matrix is not unique for a certain lattice, e.g., if $\boldsymbol{L}$ is Minkowski reduced, then so is $-\boldsymbol{L}$. However, it can be proved that there are only finitely many Minkowski reduced generator matrices for any lattice [86]. Given a generator matrix $\boldsymbol{B}$, a Minkowski reduced generator matrix $\boldsymbol{L}$, and the corresponding unimodular matrix $\boldsymbol{Z}$, can both be obtained by applying the Minkowski reduction algorithm on $\boldsymbol{B}$ [86].

Let $\boldsymbol{L}$ be a Minkowski reduced generator matrix and $\boldsymbol{G}_{\boldsymbol{L}} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$ the corresponding Minkowski reduced PQF. Condition (i) in (4.31) implies that $D_{\min}^2(\boldsymbol{L}) \geq g_{1,1}$, and since $g_{1,1} = \|\boldsymbol{l}_1\|^2$, it follows that $D_{\min}^2(\boldsymbol{L}) = g_{1,1}$, because at least $\boldsymbol{v} = (1\,0\,0\ldots0)^{\mathrm{T}}$ achieves equality. Hence, any Minkowski reduced generator matrix $\boldsymbol{L}$ contains the shortest vector in the lattice $\Lambda_{\boldsymbol{L}}$ as one of its columns. $\mathcal{M}$ is an intersection of infinitely many halfspaces, just as the Ryshkov polytope, but with different halfspaces in this case. It is easily seen that $\mathcal{M}$ corresponds to a cone in the vector space $\mathbb{R}^{N(N+1)/2}$, since if $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathcal{M}$, then $k_1\boldsymbol{G}_1 + k_2\boldsymbol{G}_2 \in \mathcal{M}$ for $k_1 \geq 0$ and $k_2 \geq 0$. We now define

Definition 9. $\mathcal{M}_\lambda = \{\boldsymbol{G} \,:\, \boldsymbol{G} \in \mathcal{M},\ g_{1,1} \geq \lambda\}$.

Hence, the Minkowski reduced PQFs in $\mathcal{M}_\lambda$ correspond to all Minkowski reduced generator matrices of lattices with a minimum distance of at least λ.

A polyhedral cone is a cone with a finite number of flat faces, and is therefore also a polytope. A fundamental result by Minkowski is [86]

Theorem 12. *$\mathcal{M}$ is a polyhedral cone in $\mathbb{R}^{N(N+1)/2}$.*

Theorem 12 is of importance later, since the polyhedral structure of $\boldsymbol{M}$ is crucial for the solvability of (4.30).

It follows from Definition 9 that $\mathcal{M}_\lambda$ is the intersection of the hyperplane $\{\boldsymbol{G} \,:\, g_{1,1} = \lambda\}$ with $\mathcal{M}$, and from Theorem 12 we conclude that $\mathcal{M}_\lambda$ contains

a finite number of vertices. The vertices are hereinafter denoted as *Minkowski extreme forms*, and the corresponding lattices as *Minkowski extreme lattices*. Minkowski extreme forms have been tabulated up to dimension 7, while for higher dimensions they are unknown since the computational complexity is too high. Compare this to perfect forms in the Ryshkov polytope, which are known up to dimension 8. This is due to the fact that enumerating perfect forms is computationally more tractable than enumerating Minkowski extreme forms [84]. Ryshkov managed to show that every perfect form is equivalent to a form lying on an extreme ray of the Minkowski reduction region $\mathcal{M}$ [87]. Cohn et al., however, showed that there are extreme rays in the Minkowski reduction region that do not contain perfect forms [88]. Thus, every vertex (perfect form) of the Ryshkov polytope $\mathcal{R}_1$ can be reduced to a vertex in $\mathcal{M}_1$, but there are some extreme rays in $\mathcal{M}_1$ which contain PQFs in $\mathcal{R}_1$ that are not vertices of $\mathcal{R}_1$. Thus, since the Minkowski reduction region is different from the Ryshkov polytope, defining our optimization problem over it can provide additional insights to the properties of the optimal solution.

4.3.2 Lattice-based problem formulation

We are now ready to reformulate (4.30) as a pure lattice optimization problem. This will, for completeness, be done over the Ryshkov polytope as well as over the Minkowski reduction region. We begin with the former. Start by factorizing $\boldsymbol{F}$ as $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{B}$, where $\boldsymbol{U}$ is an orthogonal matrix and $\boldsymbol{B}$ is any matrix such that $\boldsymbol{F}$ satisfies the trace constraint $\mathrm{tr}(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}) \leq P_0$. From lattice theory, it follows that $\boldsymbol{B}$ can be regarded as a generator matrix for a lattice $\boldsymbol{\Lambda}_{\boldsymbol{B}}$. Inserting the expression $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{B}$ into (4.30), we arrive at

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{B}} \ & \mathrm{tr}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{B}) \\ & \text{subject to} \\ & \boldsymbol{G}_{\boldsymbol{B}} \in \mathcal{R}_1, \end{aligned} \tag{4.32}$$

where $\boldsymbol{G}_{\boldsymbol{B}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$. The subscript in $\boldsymbol{G}_{\boldsymbol{B}}$ will be left out when no confusion can arise.

Let us now instead turn to the second formulation and formulate (4.30) as an optimization over the Minkowski polytope $\mathcal{M}_1$. We keep the factorization $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{B}$, but we further factorize $\boldsymbol{B}$ as $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{Z}$, where $\boldsymbol{L}$ is a Minkowski reduced basis of the lattice $\Lambda_{\boldsymbol{B}}$ and $\boldsymbol{Z}$ a unimodular matrix. This gives that the $\boldsymbol{F}$ in (4.30) can also be factorized as $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}$. Furthermore, the constraint $D^2_{\min}(\boldsymbol{B}) \geq 1$ is now equivalent to $\boldsymbol{G}_{\boldsymbol{L}} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L} \in \mathcal{M}_1$. Thus, (4.30)

can as well be formulated as

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{L},\boldsymbol{Z}} \ & \mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}) \\ & \text{subject to} \\ & \boldsymbol{G}_{\boldsymbol{L}} \in \mathcal{M}_1. \end{aligned} \tag{4.33}$$

Also for (4.33), the subscript in $\boldsymbol{G}_{\boldsymbol{L}}$ will sometimes be left out. Note that the objective functions in (4.32) and (4.33) are exactly the same, since $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{Z}$, but the optimization procedure is different for the two problems. First of all, the optimization domains are different. Secondly, the optimization in (4.32) only involves a minimization over orthogonal ($\boldsymbol{U}$) and invertible ($\boldsymbol{B}$) matrices, while (4.33) is a minimization over orthogonal, invertible ($\boldsymbol{L}$) *and* unimodular matrices ($\boldsymbol{Z}$). Hence, the methodology for solving (4.32) differs from the one solving (4.33). The formulation in (4.33) also reveals the fact that changing the basis in $\Lambda_{\boldsymbol{B}}$, i.e., varying $\boldsymbol{Z}$, only affects the transmitted power, which is not evident from the formulation in (4.32). Another advantage of the formulation in (4.33) will be revealed by Theorem 13 in Section 4.3.1. The work in [79] considered a problem formulation similar to (4.33), but without using the Minkowski reduction domain. Instead, only the objective function was studied for different lattice bases $\boldsymbol{L}$. As mentioned in Section3.5.4, both [89] and [79] made approximations to the minimum distance problem, and the hypothesis was that the densest packing lattices in high dimensions should produce large distances. However, no exact results were presented. Instead, in [79] it was just proposed that $\boldsymbol{L}$ should be a basis for the densest lattice packing, and a heuristic, iterative algorithm was given to find the optimal $\boldsymbol{Z}$. The derived precoders turn out to have good performance, however the question remains whether they indeed are optimal minimum distance precoders, and if not, how far away they are from the optimum. Thus, (4.33) was not satisfactory treated in [79]. Moreover, the results in Section 3.5.4 suggest that packing lattices are not always optimal.

Two fundamental questions arise about the problems (4.32) and (4.33): 1) Is there an explicit formula for the optimal solutions to any of the problems? 2) If there is no such formula, what is the structure of the solution and can it be found in a simple way for any channel outcome $\boldsymbol{S}$?

Before answering these questions, we look at some classical lattice problems and tools for solving lattice optimization problems. Our motivation for surveying these well known problems is that the minimum distance problem studied in this thesis is tightly connected to them, and has in principle the same structure in its solution as some of the classical problems.

4.3.3 Classical lattice problems

There are many optimization problems that can be interpreted as optimization over lattices. A famous one is finding the *densest lattice packing* of spheres in an N-dimensional space, corresponding to the following optimization

$$\begin{aligned} &\min_{\boldsymbol{L}} \operatorname{Vol}(\boldsymbol{L}) \\ &\text{subject to } D^2_{\min}(\boldsymbol{L}) \geq 1, \end{aligned} \tag{4.34}$$

i.e., to find, among all lattices with fixed minimum distance, the lattice with the minimal volume. The dual of this problem is to find the lattice maximizing the volume of the sphere *encompassed* by its Voronoi region; this is known as maximizing the *covering* of the lattice. Mathematically, it corresponds to the following optimization

$$\begin{aligned} &\min_{\boldsymbol{L}} \max_{\boldsymbol{w} \in \mathcal{V}(\boldsymbol{L})} \|\boldsymbol{w}\| \\ &\text{subject to} \\ &D^2_{\min}(\boldsymbol{L}) \geq 1. \end{aligned} \tag{4.35}$$

The general solutions of these problems remain unknown as of today. However, for small enough dimensions, solutions are known. In two dimensions, it turns out that the hexagonal lattice solves both of these problems; this fact was shown for (4.34) by Lagrange in 1801 [90], and for (4.35) by Kershner in 1934 [74]. The packing problem has been solved for $N \leq 9$ and $N = 24$, while it is unsolved for all other N. For the covering problem, the solution is known for $N \leq 5$. Although the packing problem is unsolved in general, it is known that the optimal lattice must be a perfect lattice which also immediately implies that it is attained at a Minkowski extreme lattice. Hence, finding the densest lattice packing in any dimension N amounts to traversing the non-isometric vertices in $\mathcal{R}_1$, or traversing the vertices in $\mathcal{M}_1$. Although the former is computationally more feasible, traversing the non-isometric perfect forms also becomes computationally inefficient for higher dimensions. Despite the computational bottleneck, it is known that (4.34) and (4.35) are both discrete optimization problems rather than continuous ones.

To show that the solution of (4.34) is achieved by a perfect lattice (or a Minkowski extreme lattice), it suffices to show that $\operatorname{Vol}(\boldsymbol{L})$ is a strictly concave function over $\mathcal{S}^{N\times N}_{\succ 0}$. Since $\mathcal{S}^{N\times N}_{\succ 0}$ contains the polytopes $\mathcal{R}_1$ and $\mathcal{M}_1$, this therefore implies that $\operatorname{Vol}(\boldsymbol{L})$ is concave over both $\mathcal{R}_1$ and $\mathcal{M}_1$. Therefore, the solution to (4.34) is attained at the vertices of these polytopes, i.e., at the perfect lattices (vertices of $\mathcal{R}_1$) and the Minkowski extreme lattices (vertices of $\mathcal{M}_1$). Hence, *concavity* of the objective function is enough to conclude that

perfect lattices solve a given lattice optimization problem. The concavity of $\det(\boldsymbol{G})^{1/N}$ over $\mathcal{S}_{\succ 0}^{N\times N}$ was shown by Minkowski [86].

Next, we show that the objective functions in (4.32) and (4.34) are of different nature. Recall that (4.32) is the minimum distance optimization problem, while (4.34) is the optimal lattice packing problem. The orthogonal matrix $\boldsymbol{U}$ minimizing the objective function in (4.32) is given by Theorem 7, and equals the left orthogonal matrix in the SVD decomposition of $\boldsymbol{B}$. Inserting this $\boldsymbol{U}$ into the objective function in (4.32) gives an optimization problem as in (3.12), but here in terms of a lattice optimization problem

$$\begin{aligned} &\min_{\boldsymbol{G}} \sum_{j=1}^{N} \omega_j(\boldsymbol{G_B})/s_{j,j}^2 \\ &\text{subject to } \boldsymbol{G_B} \in \mathcal{R}_1, \end{aligned} \tag{4.36}$$

where $\omega_j(\boldsymbol{G_B})$ is the j:th largest eigenvalue of $\boldsymbol{G_B}$ and $s_{j,j}$ is the j:th largest diagonal element in $\boldsymbol{S}$. The optimization in (4.34) can be performed over the Ryshkov polytope, with the objective function $\sqrt[N]{\det(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{B})} = \sqrt[N]{\det(\boldsymbol{S}^{-2})\det(\boldsymbol{G_B})}$. The matrix $\boldsymbol{S}$ can be regarded as a constant and does not impact the optimization. It further holds that

$$\sqrt[N]{\det(\boldsymbol{S}^{-2})\det(\boldsymbol{G_B})} = \sqrt[N]{\prod_{j=1}^{N} \omega_j(\boldsymbol{G_B})/s_{j,j}^2}.$$

Hence, (4.34) minimizes the N:th root of the product of the eigenvalues of $\boldsymbol{G_B}$ over $\mathcal{R}_1$, while (4.36) minimizes a weighted sum of them. Due to the arithmetic-geometric mean (AM-GM) inequality, we have that

$$\sum_{j=1}^{N} \omega_j(\boldsymbol{G})/s_{j,j}^2 \geq N \sqrt[N]{\prod_{j=1}^{N} \omega_j(\boldsymbol{G})/s_{j,j}^2},$$

which shows that the $\boldsymbol{L}$ solving (4.34) is only minimizing the lower bound to the objective function in (4.36), thus not guaranteeing that it is the optimum to (4.36)[4] Hence, although (4.32) and (4.34) have the same optimization domain, (4.32) posseses a different objective function than (4.34), and is thus a different lattice optimization problem.

[4]This same reasoning is used in [79] in order to propose densest lattices as good candidates for providing a large minimum distance.

4.3.4 Optimal lattice structure

This section will prove the concavity of the objective functions in (4.32) and (4.33), respectively. We start by proving the concavity of the objective function in (4.33) over $\mathcal{S}_{\succ 0}^{N\times N}$, for any given $\boldsymbol{S}$ and $\boldsymbol{Z}$ matrix. Define

$$f(\boldsymbol{L},\boldsymbol{Z}) \stackrel{\triangle}{=} \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}),$$

which is the objective function in (4.33) without the minimization over $\boldsymbol{L}$ and $\boldsymbol{Z}$. Observe also that $f(\boldsymbol{B},\boldsymbol{I}_{N\times N})$ is the objective function in (4.32) without the minimization over $\boldsymbol{B}$. We now show

Theorem 13. *For a fixed $\boldsymbol{Z}$, $f(\boldsymbol{L},\boldsymbol{Z})$ is concave over $\mathcal{S}_{\succ 0}^{N\times N}$ with respect to $\boldsymbol{G}_{\boldsymbol{L}}$.*

Proof. Write $\min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}) = \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})$. Let $\boldsymbol{L}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^{\mathrm{T}}$ be the eigenvalue decomposition of $\boldsymbol{L}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}$, where $\boldsymbol{Q}$ is the orthogonal matrix. Now note that

$$\begin{aligned}
f(\boldsymbol{L},\boldsymbol{Z}) &= \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}) \\
&= \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{Q}\boldsymbol{D}^{\boldsymbol{2}}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}) \\
&= \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{D}^{2}\boldsymbol{Q}\boldsymbol{U}^{\mathrm{T}}) \\
&= \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z}\boldsymbol{U}^{\mathrm{T}}) \\
&= \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}}\boldsymbol{Z}\boldsymbol{U}^{\mathrm{T}}).
\end{aligned}$$

Hence

$$\begin{aligned}
f(\boldsymbol{L},\boldsymbol{Z}) &= h(\boldsymbol{G}_{\boldsymbol{L}},\boldsymbol{Z}) \\
&= \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}}\boldsymbol{Z}\boldsymbol{U}^{\mathrm{T}}).
\end{aligned}$$

Now it follows that for positive semidefinite $\boldsymbol{G}_1$, $\boldsymbol{G}_2$ and $0 \le \gamma \le 1$,

$$\begin{aligned}
h(\gamma\boldsymbol{G}_1 + (1-\gamma)\boldsymbol{G}_2,\boldsymbol{Z}) &= \min_{\boldsymbol{U}} \gamma\mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{G}_1\boldsymbol{U}^{\mathrm{T}}) + (1-\gamma)\mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{G}_2\boldsymbol{U}^{\mathrm{T}}) \\
&\ge \gamma \min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{G}_1\boldsymbol{U}^{\mathrm{T}}) + (1-\gamma)\min_{\boldsymbol{U}} \mathrm{tr}(\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{G}_2\boldsymbol{U}^{\mathrm{T}}) \\
&= \gamma h(\boldsymbol{G}_1) + (1-\gamma)h(\boldsymbol{G}_2),
\end{aligned}$$

which shows that $h(\boldsymbol{G}_{\boldsymbol{L}},\boldsymbol{Z})$, and thus also $f(\boldsymbol{L},\boldsymbol{Z})$, are concave over $\mathcal{S}_{\succ 0}^{N\times N}$ with respect to $\boldsymbol{G}_{\boldsymbol{L}}$. □

An immediate corollary of Theorem 13 is

Corollary 3. *$f(\boldsymbol{L}, \boldsymbol{Z})$ is concave over $\mathcal{S}_{\succ 0}^{N\times N}$ with respect to the Gram matrix $\boldsymbol{G_B} = \boldsymbol{G_{LZ}} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{LZ}$.*

Proof. Let $\boldsymbol{L} = \boldsymbol{B}$, $\boldsymbol{Z} = \boldsymbol{I}_{N\times N}$ and apply Theorem 13. □

Taken together, Theorem 13 and Corollary 1 show that the objective functions in (4.32) and (4.33), respectively, are both concave over their corresponding domains. This immediately implies that the solution to (4.32) is a perfect lattice, and the solution to (4.33) is a Minkowski extreme lattice. Exactly which perfect lattice/Minkowski extreme lattice that solves (4.30) depends of course on the channel outcome $\boldsymbol{S}$, and an algorithm is given in Section 4.3.6 that enumerates all possible perfect forms solving (4.30) for a specific $\boldsymbol{S}$. However, since there are finitely many perfect lattices/Minkowski extreme lattices in N dimensions, we know that there are finitely many different lattices solving the problem for all $\boldsymbol{S}$. This answers our second question posed in Section 4.3.2: The optimal $\boldsymbol{G_B}$ in (4.32) is a vertex of the polytope $\mathcal{R}_1$, and the optimal $\boldsymbol{G_L}$ solving (4.33) corresponds to a vertex in $\mathcal{M}_1$. Hence, the solution to (4.30) does not depend continuously on $\boldsymbol{S}$, instead it changes in a discrete fashion when $\boldsymbol{S}$ is varied continuously. Relating to the first question in Section 4.3.2, this result implies that an explicit formula for the solution of (4.30) seems out of reach, since such a formula does not exist for (4.34) whose set of possible solutions is a subset of the set of possible solutions to (4.30). Altogether, a previously unknown result is revealed: There are finitely many lattices that can solve the minimum distance optimization problem in (4.30), and they can be enumerated offline.

Theorem 13 also reveals that for *any* given $\boldsymbol{Z}$ matrix, the optimal solution to (4.33) occurs at a Minkowski extreme lattice. Thus, given any $\boldsymbol{Z}$, the optimal $\boldsymbol{L}$ that builds up $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{ULZ}$ in (4.30) is a Minkowski extreme lattice. This fact will be used in Section 4.3.6 to develop a good suboptimal precoder construction. Hence, the problem formulation in (4.33) provides additional information about the behavior of (4.30), not present in (4.32): This is the main reason for introducing (4.33).

We can already at this stage deduce several interesting conclusions from the result in Theorem 13. For up to three dimensions, there is only one non-isometric perfect lattice in each dimension: For $N = 2$ it is the hexagonal lattice and in $N = 3$ it is the face-centered cubic lattice. Since these are the only non-isometric perfect lattices in these dimensions, they also solve (4.34), and thus the proposition in [79] to use densest lattice packings in (4.33) is optimal for these dimensions. However, when $N = 4$ there are two non-isometric lattices: The checkerboard lattice D_4 and the root lattice A_4 [74].

It will be demonstrated in Section 4.3.8 that *both* of these lattices occur as solutions to (4.32) for different $\boldsymbol{S}$, so the constructions in [79] are suboptimal for $N = 4$. Another interesting consequence of Theorem 13 is that the main result in Theorem 8, which shows that the hexagonal lattice is optimal in two dimensions, now follows immediately from Theorem 13. However, as will be explained in Section 4.4, the results in this section do not cover the result in Theorem 9.

Note that it is now an easy task to construct the optimal $\boldsymbol{F}$ in (4.30), once the $\boldsymbol{G_B}$ solving (4.32) is known. Let $\boldsymbol{G}_{\text{opt}} = \boldsymbol{B}_{\text{opt}}^{\text{T}}\boldsymbol{B}_{\text{opt}}$ denote the optimal form and $\boldsymbol{G}_{\text{opt}} = \boldsymbol{Q}_{\text{opt}}\boldsymbol{D}_{\text{opt}}\boldsymbol{Q}_{\text{opt}}^{\text{T}}$ be its eigenvalue decomposition. Since the optimal $\boldsymbol{U}$ in $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{B}$ is equal to the left orthogonal matrix in the SVD decomposition of $\boldsymbol{B}$, it follows that the optimal $\boldsymbol{F}$ can be constructed as

$$\boldsymbol{F}_{\text{opt}} = \boldsymbol{S}^{-1}\sqrt{\boldsymbol{D}_{\text{opt}}}\boldsymbol{U}_{\text{opt}}^{\text{T}}. \tag{4.37}$$

To summarize, the following knowledge is at hand about the solution to the original problem in (4.30). We have shown that (4.30) is equivalent to both (4.32) and (4.33). Theorem 13 then shows that the $\boldsymbol{L}$ matrix solving (4.33), for *any* invertible $\boldsymbol{S}$, gives rise to a Gram matrix $\boldsymbol{G} = \boldsymbol{L}^{\text{T}}\boldsymbol{L}$ that corresponds to a vertex in the polytope $\mathcal{M}_1$. Since $\mathcal{M}_1$ has a finite number of vertices for any dimension N, and is independent of the matrix $\boldsymbol{S}$, it holds that there are finitely many $\boldsymbol{L}$ matrices (up to rotation) that are candidates to solving (4.33) for any given $\boldsymbol{S}$. Once the optimal $\boldsymbol{L}_{\text{opt}}$ is known (up to rotation), it remains to find the optimal unimodular matrix $\boldsymbol{Z}_{\text{opt}}$ in (4.33) and then to construct the optimal precoder $\boldsymbol{F}_{\text{opt}}$ from (4.37), where $\boldsymbol{B}_{\text{opt}} = \boldsymbol{L}_{\text{opt}}\boldsymbol{Z}_{\text{opt}}$. To find the optimal $\boldsymbol{Z}$, we need to perform a search over unimodular matrices, which can be simplified if good bounds on the optimum solution to (4.33) are known. These bounds will be developed in Section 4.3.5.

When it comes to the equivalent problem formulation in (4.32), Corollary 3 shows that the $\boldsymbol{B}$ matrix solving (4.32) for any given invertible $\boldsymbol{S}$, is such that it produces a Gram matrix $\boldsymbol{G} = \boldsymbol{B}^{\text{T}}\boldsymbol{B}$ that is one of the vertices in the polytope $\mathcal{R}_1$. Given the optimal $\boldsymbol{G}$ in $\mathcal{R}_1$, the optimal precoder is obtained through (4.37). The $\mathcal{R}_1$ polytope contains infinitely many vertices, and it is known that each vertex is isometric to some vertex in $\mathcal{M}_1$. Since (4.32) is connected to (4.33) through the factorization $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{Z}$, it holds that if $\boldsymbol{G_B} = \boldsymbol{B}^{\text{T}}\boldsymbol{B} = \boldsymbol{Z}^{\text{T}}\boldsymbol{L}^{\text{T}}\boldsymbol{L}\boldsymbol{Z}$ is a vertex in $\mathcal{R}_1$, then $\boldsymbol{G_L} = \boldsymbol{L}^{\text{T}}\boldsymbol{L}$ is a vertex in $\mathcal{M}_1$. Hence, traversing the different vertices in $\mathcal{R}_1$ is equivalent to a joint enumeration of some of the vertices in $\mathcal{M}_1$ and different unimodular matrices $\boldsymbol{Z}$. However, it is instead possible to directly enumerate perfect forms by formulating an algorithm working over $\mathcal{R}_1$. Again, bounds are needed in order

to restrict the amount of vertices to traverse, and they will be presented in the next section.

4.3.5 Bounds on the optimal solution

We start by deriving lower and upper bounds to $\mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z})$, which is the objective function in (4.33). The upper bound presented here improves significantly upon the upper bound presented in [79]. From these bounds, we are able to derive further bounds that aid in restricting the search space for the algorithms that find the optimal precoder, which are introduced in Section 4.3.6.

Theorem 14. *The following lower bound holds for the optimal solution to* (4.33)

$$\min_{\boldsymbol{U},\boldsymbol{L},\boldsymbol{Z}} \mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}) \geq N\sqrt[N/2]{\det(\boldsymbol{L})/\det(\boldsymbol{S})}. \tag{4.38}$$

Proof. Dropping the integer-valued constraint on $\boldsymbol{Z}$, while keeping the determinant constraint $\det(\boldsymbol{Z}) = \pm 1$, we apply the method of Lagrange multipliers to find first order optimality conditions. Let $\boldsymbol{M} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}$. The optimal $\boldsymbol{Z}_o$ must satisfy

$$\frac{\partial \mathrm{tr}(\boldsymbol{Z}_o^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}_o)}{\partial \boldsymbol{Z}_o} = \gamma \frac{\det(\boldsymbol{Z}_o) - 1}{\partial \boldsymbol{Z}_o} \Rightarrow 2\boldsymbol{Z}_o^{\mathrm{T}}\boldsymbol{M} = \gamma \left(\boldsymbol{Z}_o^{\mathrm{T}}\right)^{-1}, \tag{4.39}$$

where $\gamma \in \mathbb{R}$. Taking determinants on both sides, and making use of $\det(\boldsymbol{Z}_o) = \pm 1$, we get $\gamma = 2\sqrt[N]{\det(\boldsymbol{M})}$. Inserting this γ into (4.39) and multiplying both sides of the equation with $\boldsymbol{Z}_o^{\mathrm{T}}$, we arrive at $\boldsymbol{Z}_o^{\mathrm{T}}\boldsymbol{M}\boldsymbol{Z}_o = \sqrt[N]{\det(\boldsymbol{M})}\boldsymbol{I}_N$. Hence, for this $\boldsymbol{Z}_o$, we get

$$\begin{aligned} \mathrm{tr}(\boldsymbol{Z}_o^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}_o) &= \mathrm{tr}(\boldsymbol{I}_N)\sqrt[N]{\det(\boldsymbol{M})} \\ &= N\sqrt[N]{\det(\boldsymbol{M})}. \end{aligned}$$

Since this expression is independent of $\boldsymbol{U}$, it is a lower bound to the objective function for a fixed $\boldsymbol{L}$. Expressing $\det(\boldsymbol{M}) = \det(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{L}) = \det^2(\boldsymbol{L})/\det^2(\boldsymbol{S})$, we arrive at the lower bound in (4.38). □

This bound was also reported in [79], but derived in a different way, by using the AM-GM inequality and Hadamard's inequality. The approach presented here shows that this lower bound corresponds to the optimal *real-valued* unimodular $\boldsymbol{Z}$.

Next, we derive an upper bound on $\min_{\boldsymbol{U},\boldsymbol{L},\boldsymbol{Z}} \mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z})$

Theorem 15. *If* $D^2_{\min}(\boldsymbol{L}) = 1$*, then*

$$\min_{\boldsymbol{U},\boldsymbol{L},\boldsymbol{Z}} \operatorname{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}) \leq N\sqrt[N/2]{1/\det(\boldsymbol{S})}. \tag{4.40}$$

Proof. Let $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{L}\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R}$ denote the QR-decomposition of the received lattice. It holds that $D^2_{\min}(\boldsymbol{B}) \geq \min_{1\leq i\leq N} |r_{i,i}|^2$ [91]. In [92], an orthogonal precoder matrix $\boldsymbol{F}_{\mathrm{gmd}}$ (geometric mean precoder) and an orthogonal receiver matrix $\boldsymbol{W}_{\mathrm{gmd}}$ were constructed, such that in the QR decomposition of $\boldsymbol{B}_{\mathrm{gmd}} = \boldsymbol{W}_{\mathrm{gmd}}\boldsymbol{S}\boldsymbol{F}_{\mathrm{gmd}}$, all diagonal elements of $\boldsymbol{R}$ equal $\sqrt[N]{\det(S)}$. Hence, in essence, the precoder $\boldsymbol{F}_{\mathrm{gmd}}$ together with the rotation $\boldsymbol{W}_{\mathrm{gmd}}$ at the receiver, produces a lattice with maximal lower bound on $D^2_{\min}$. This value is equal to the geometric mean of its singular values, thereby its name the "geometric mean precoder". It is clear that $D^2_{\min}(\boldsymbol{B}_{\mathrm{gmd}}) \geq \sqrt[N]{\det(S)}$, and since $\boldsymbol{F}_{\mathrm{gmd}}$ is orthogonal, $\operatorname{tr}(\boldsymbol{F}^{\mathrm{T}}_{\mathrm{gmd}}\boldsymbol{F}_{\mathrm{gmd}}) = N$. Hence $d^2_{\min}(\boldsymbol{S},\boldsymbol{F}_{\mathrm{gmd}}) = D^2_{\min}(\boldsymbol{B}_{\mathrm{gmd}})/\operatorname{tr}(\boldsymbol{F}^{\mathrm{T}}_{\mathrm{gmd}}\boldsymbol{F}_{\mathrm{gmd}}) \geq \sqrt[N/2]{\det(\boldsymbol{S})}/N$. Now it follows that for any precoder $\boldsymbol{F}$ with higher $d^2_{\min}(\boldsymbol{S},\boldsymbol{F})$ than $d^2_{\min}(\boldsymbol{S},\boldsymbol{F}_{\mathrm{gmd}})$, $d^2_{\min}(\boldsymbol{S},\boldsymbol{F}) \geq (\det(\boldsymbol{S}))^{2/N}/N$. Hence, this gives an upper bound on $\operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^{\mathrm{T}})$, $\operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^{\mathrm{T}}) \leq N\sqrt[N/2]{1/\det(\boldsymbol{S})}$. Writing $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}$, we get the upper bound in (4.40). □

Combining Theorem 14 and 15, we have the following bounds

$$N(\det(\boldsymbol{L})/\det(\boldsymbol{S}))^{2/N} \leq \operatorname{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}) \leq N(1/\det(\boldsymbol{S}))^{2/N}. \tag{4.41}$$

Recall that $d^2_{\min}(\boldsymbol{S},\boldsymbol{F})$ is the ratio of the minimum distance to the energy of the precoder. In terms of this ratio, the bounds in (4.41) translate into

$$(\det(\boldsymbol{S})/\det(\boldsymbol{L}))^{2/N}/N \geq d^2_{\min}(\boldsymbol{S},\boldsymbol{F}) \geq (\det(\boldsymbol{S}))^{2/N}/N. \tag{4.42}$$

Note that the $\boldsymbol{L}$ in (4.41) and (4.42) is such that $D^2_{\min}(\boldsymbol{L}) = 1$. Also, these bounds hold for *any* precoder $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}$ that improves upon $\boldsymbol{F}_{\mathrm{gmd}}$. It is readily seen that the ratio between the upper bound and the lower bound in (4.41) is $(1/\det(\boldsymbol{L}))^{2/N}$. Hence, for a fixed dimension, the optimum ratio $d^2_{\min}(\boldsymbol{S}\boldsymbol{F}_{\mathrm{opt}})/(\det(\boldsymbol{S}))^{2/N}$ is always smaller than $(1/\det(\boldsymbol{L}))^{2/N}$, *independently* of the channel $\boldsymbol{S}$. For example, when $N = 2$, the optimal lattice is the hexagonal lattice $\boldsymbol{L}_{\mathrm{hex}}$ and when $D^2_{\min}(\boldsymbol{L}_{\mathrm{hex}}) = 1$, $\det(\boldsymbol{L}_{\mathrm{hex}}) = \sqrt{3/4}$. The ratio between the upper bound and lower bound in (4.42) is then $\sqrt{4/3} \approx 1.16$, hence the optimal $d^2_{\min}(\boldsymbol{S}\boldsymbol{F}_{\mathrm{opt}})$ in two dimensions is at most 16% better than the lower bound $d^2_{\min}(\boldsymbol{S}\boldsymbol{F}_{\mathrm{gmd}}) = \det(\boldsymbol{S})/2$. For $N = 3$, the optimal lattice is the face-centered cubic lattice $\boldsymbol{L}_{A_3}$ that has a volume of $\det(\boldsymbol{L}_{A_3}) = 1/2$ when $D^2_{\min}(\boldsymbol{L}_{A_3}) = 1$. In this case, the ratio between the bounds in (4.42) is $2^{1/3} \approx 1.26$; hence, the performance of the optimal precoder is at most 26% better than for $\boldsymbol{F}_{\mathrm{gmd}}$. It is

worthwile to observe that the ratio between the bounds, for optimal packing lattices $\boldsymbol{L}$, equals Hermite's constant $\delta_N \triangleq \max_{\boldsymbol{L}} D^2_{\min}(\boldsymbol{L})/\mathrm{Vol}(\boldsymbol{L})^{2/N}$ [85]. Thus, the ratio of the bounds is upper bounded by Hermite's constant. Hermite's constant is the ratio between the constraint function and the objective function in (4.34), and is therefore an optimization problem equivalent to (4.34). The following upper and lower bounds are known for δ_N [74, 93]

$$\frac{N}{2\pi e} + \frac{\log(\pi N)}{2\pi e} + c_{N,1} \leq \delta_N \leq \frac{1.744N}{2\pi e}(1 + c_{N,2}),$$

where $c_{N,1}$ and $c_{N,2}$ are constants depending on the dimension N. From this it follows that δ_N grows linearly with the dimension N, and thus the ratio of our bounds grows at most linearly with N. This can be compared to the bounds in [79], where the ratio between the upper bound and lower bound contains the exponential factor $2^{N/2}$. Thus, the improvement in the upper bound is significant. Note also that $\boldsymbol{F}_{\mathrm{gmd}}$ operates above the lower bound in (4.42), and due to its good SER performance as reported in [92], it can serve as a basis for developing a suboptimal $\boldsymbol{Z}$ matrix to (4.33). This will be presented in Section 4.3.6.

As discussed in the first paragraph of Section 4.3.4, a closed form solution to (4.30) seems out of reach. We are thus interested in an algorithm that can find the optimal $\boldsymbol{Z}$ in (4.33), or an algorithm to find the optimal $\boldsymbol{G_B}$ in (4.32). In order to do so, it is desirable to first have some bounds on the $\boldsymbol{Z}$ matrix or some quantity depending on it, in order to restrict the search space. From this perspective, we develop an upper bound on $\mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z})$.

Theorem 16. *With $D^2_{\min}(\boldsymbol{L}) = 1$, the following upper bound holds*

$$\min_{\boldsymbol{L},\boldsymbol{Z}} \mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z}) \leq N\left(\frac{s_{1,1}}{\sqrt[n]{\det(\boldsymbol{S})}}\right)^2. \tag{4.43}$$

Proof. Let $\boldsymbol{G_L} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z}$. Inserting the optimal $\boldsymbol{U}$ into (4.41), we arrive at the upper bound

$$\mathrm{tr}(\Omega(\boldsymbol{G_L})\boldsymbol{S}^{-2}) \leq N(1/\det(\boldsymbol{S}))^{2/N}, \tag{4.44}$$

where $\Omega(\boldsymbol{G_L})$ is the diagonal matrix containing the eigenvalues $\omega_j(\boldsymbol{G_L})$ of $\boldsymbol{G_L}$. Since the eigenvalues $\omega_j(\boldsymbol{G_L})$ are sorted in opposite order to $s^{-2}_{j,j}$, and $s^{-2}_{1,1} \leq \ldots \leq s^{-2}_{N,N}$, we have the inequality

$$\mathrm{tr}(\Omega(\boldsymbol{G_L})\boldsymbol{S}^{-2}) \geq \frac{\mathrm{tr}(\Omega(\boldsymbol{G_L}))}{s^2_{1,1}},$$

which gives us the upper bound in (4.43) □

Geometrically, the inequality in (4.43) implies that the lattice vectors of the optimal lattice $\boldsymbol{L}_{\text{opt}}\boldsymbol{Z}_{\text{opt}}$ must have a bounded length. Using the trace inequality [94]

$$\omega_N(\boldsymbol{G}_{\boldsymbol{L}_{\text{opt}}})\text{tr}(\boldsymbol{Z}_{\text{opt}}\boldsymbol{Z}_{\text{opt}}^{\text{T}}) \leq \text{tr}(\boldsymbol{Z}_{\text{opt}}^{\text{T}}\boldsymbol{L}_{\text{opt}}^{\text{T}}\boldsymbol{L}_{\text{opt}}\boldsymbol{Z}_{\text{opt}}) \leq \omega_1(\boldsymbol{G}_{\boldsymbol{L}_{\text{opt}}})\text{tr}(\boldsymbol{Z}_{\text{opt}}\boldsymbol{Z}_{\text{opt}}^{\text{T}}),$$

we also have the following upper bound for $\boldsymbol{Z}_{\text{opt}}$

$$\text{tr}(\boldsymbol{Z}_{\text{opt}}\boldsymbol{Z}_{\text{opt}}^{\text{T}}) \leq \frac{N}{\omega_N(\boldsymbol{G}_{\boldsymbol{L}_{\text{opt}}})}\left(\frac{s_{1,1}}{\sqrt[n]{\det(\boldsymbol{S})}}\right)^2. \tag{4.45}$$

In terms of $\boldsymbol{G}_{\boldsymbol{B}} = \boldsymbol{Z}^{\text{T}}\boldsymbol{L}^{\text{T}}\boldsymbol{L}\boldsymbol{Z} = \boldsymbol{B}^{\text{T}}\boldsymbol{B}$, the upper bound in (4.43) is

$$\text{tr}(\boldsymbol{G}_{\boldsymbol{B}}) \leq N\left(\frac{s_{1,1}}{\sqrt[n]{\det(\boldsymbol{S})}}\right)^2. \tag{4.46}$$

Let $\text{ub}(\boldsymbol{S})$ denote the upper bound in (4.46). Hence, the optimal $\boldsymbol{G}_{\boldsymbol{B}}$ in the Ryshkov polytope that solves (4.32) is one of the vertices of the finite, bounded polytope $\mathcal{R}_1 \cap \{\boldsymbol{G}_{\boldsymbol{B}} : \text{tr}(\boldsymbol{G}_{\boldsymbol{B}}) \leq \text{ub}(\boldsymbol{S})\}$.

4.3.6 Numerical methods for solving (4.32) and (4.33)

In this section, we present algorithmical approaches to solve (4.32) and (4.33). Although (4.32) can be solved by enumerating all vertices in the polytope $\mathcal{R}_1 \cap \{\boldsymbol{G} : \text{tr}(\boldsymbol{G}) \leq \text{ub}(\boldsymbol{S})\}$, the methodology for solving (4.33) will provide another interesting observation. Additionally, the problem formulation in (4.33) gives novel insight into an efficient suboptimal precoder construction, formulated in Section 4.3.7, that is not present in the formulation in (4.32). First, we discuss a method to solve (4.33), then we discuss the solution to (4.32).

Finding the solution to (4.33)

To find the solution to (4.33), one needs to tabulate Minkowski extreme lattices in N dimensions. Unfortunately, it turns out to be more complex to enumerate Minkowski extreme lattices than perfect forms [84]. Note, however, that only those Minkowski extreme lattices that correspond to perfect forms have to be known. Namely, once all non-isometric perfect forms have been tabulated in N dimensions, the fact that each perfect form (vertex) in $\mathcal{R}_1$ is equivalent to a Minkowski extreme form (vertex) in $\mathcal{M}_1$, implies that the Minkowski extreme lattices solving (4.33) are the ones corresponding to the non-isometric perfect forms. Therefore, it is not necessary to know *all* the Minkowski extreme

lattices in N dimensions in order to solve (4.33), only those corresponding to perfect forms are needed. However, when constructing a good suboptimal solution to (4.33), presented in Section 4.3.7, it is necessary to know all the Minkowski extreme lattices to obtain the best suboptimal construction. The smallest eigenvalue $\omega_N(\boldsymbol{G_L})$ in (4.45) is non-zero for all the Minkowski extreme lattices $\boldsymbol{L}$ (candidates for the optimum), and thus the bound in (4.45) is well-defined. A geometrical interpretation is that this inequality bounds the squared lengths sum of the basis vectors in the integer lattice $\mathbb{Z}^N$, where the basis vectors are now the rows of $\boldsymbol{Z}$. Thus, finding the optimal $\boldsymbol{Z}$ can be regarded as searching for basis vectors inside a sphere of a certain radius. If one has a priori knowledge about the maximum ratio $s_{1,1}/\det(\boldsymbol{S})$, an off-line, one-shot algorithm can be formulated that searches for unimodular $\boldsymbol{Z}$ inside the largest sphere, corresponding to the Minkowski extreme lattice $\boldsymbol{L}$ with smallest $\omega_N(\boldsymbol{G_L})$ and the channel $\boldsymbol{S}$ with largest upper bound in (4.45). This sphere certainly includes the optimal $\boldsymbol{Z}_{\text{opt}}$ corresponding to the optimal Minkowski extreme lattice $\boldsymbol{L}_{\text{opt}}$ for any channel that can occur. A large codebook of matrices $\boldsymbol{LZ}$ can then be constructed off-line, by matrix multiplication of each encountered $\boldsymbol{Z}$ in the sphere with the different Minkowski extreme lattices and storing the resulting matrices into the codebook. To then find the optimal precoder $\boldsymbol{F} = \boldsymbol{S}^{-1}\boldsymbol{ULZ}$ online for a certain $\boldsymbol{S}$, one simply goes through every element $\boldsymbol{LZ}$ in the codebook, and constructs $\boldsymbol{F}$ by using the optimal $\boldsymbol{U}$.

The outlined method to solve (4.33) includes the following steps: 1) Find all Minkowski extreme lattices that correspond to non-isometric perfect forms. This is accomplished by applying Voronoi's algorithm to enumerate non-isometric perfect forms [95], and then applying the Minkowski reduction algorithm to the obtained perfect lattices. 2) Enumerate all unimodular $\boldsymbol{Z}$ satisfying the bounds in (4.45). There are specialized algorithms for this task [96].

Finding the solution to (4.32)

As described in Section 4.3.5, the optimal $\boldsymbol{G}$ is one of the vertices in the polytope $\mathcal{R}_1 \cap \{\boldsymbol{G} : \text{tr}(\boldsymbol{G}) \leq \text{ub}(\boldsymbol{S})\}$. Hence, one method to find the optimum is to directly enumerate all the perfect forms inside the polytope. A finite codebook can be constructed off-line if a priori knowledge of the upper bound in (4.46) is available. A method to enumerate perfect forms is via Voronoi's algorithm [95]. It enumerates perfect forms and stops when all non-isometric forms have been found. As mentioned, for today's computers, it is only usable up to 8 dimensions due to the large number of edges in the Ryshkov polytope in higher dimensions. We need to slightly modify the classical Voronoi's algorithm, by changing its stopping condition. Since we are interested in forms that are

isometric, our stopping condition is based on the upper bound in (4.46).

An exclusion criteria for vertices can be formulated, which helps in reducing the size of the final codebook. From the problem formulation in (4.36), we see that vertices with eigenvalues that *majorize* the eigenvalues of some other vertex, can never solve (4.36). By majorization, we mean the following. Let $\boldsymbol{a} = \{a_1, \ldots, a_N\}$, $a_1 \leq \ldots \leq a_N$, and $\boldsymbol{b} = \{b_1, \ldots, b_N\}$, $b_1 \leq \ldots \leq b_N$, be two sequences of length N sorted in ascending order. If $\boldsymbol{a}$ is majorized by $\boldsymbol{b}$, denotes as $\boldsymbol{a} \preceq \boldsymbol{b}$, then the following inequalities hold

$$\sum_{i=1}^{k} a_i \leq \sum_{i=1}^{k} b_i, \quad k = 1, \ldots, N. \tag{4.47}$$

Clearly, majorization induces a partial order on the set of sequences. Now, the objective function in (4.36) can be written in terms of partials sum. Denote $f_j \triangleq 1/s_{j,j}^2$, $j = 1, \ldots, N$, with $f_0 \triangleq 0$. Hence, $f_0 \leq f_1 \leq \ldots \leq f_N$. The objective function in (4.36) is

$$\sum_{j=1}^{N} \omega_j(\boldsymbol{G_B})/s_{j,j}^2 = \sum_{j=1}^{N} (f_j - f_{j-1}) \sum_{k=j}^{N} \omega_j(\boldsymbol{G_B}). \tag{4.48}$$

Each term $f_j - f_{j-1}$ in (4.48) is non-negative and $\sum_{k=j}^{N} \omega_j(\boldsymbol{G_B})$ is the partial sum of the eigenvalues of $\boldsymbol{G_B}$. It is clear from this formulation that if $\boldsymbol{G'_B}$ is another vertex with eigenvalues $\omega_j(\boldsymbol{G'_B})$ that are majorizing $\omega_j(\boldsymbol{G_B})$, then $\boldsymbol{G_B}$ can never produce a smaller value of the objective function in (4.48) than $\boldsymbol{G'_B}$ for *any* realization of $\{1/s_{j,j}^2\}$. Thus, $\boldsymbol{G'_B}$ does not have to be included in the codebook. However, the Voronoi algorithm still needs to traverse it, since other forms that solve (4.36) for a certain $\boldsymbol{S}$ might be reachable from it.

The algorithm needs an initial perfect form as starting position, and a good starting point is the root lattice A_N [84]. The following notation is used in the algorithm. $\text{Min}(\boldsymbol{G})$ denotes the set of minimum vectors of $\boldsymbol{G}$, i.e., the set $\{\boldsymbol{x} : \boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{x} = 1\}$ and $\boldsymbol{G}[\boldsymbol{x}] \triangleq \boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{x}$. The algorithm is summarized by the pseudo-code in Table 4.1. The only difference between Algorithm in 4.1 and the Voronoi algorithm presented in [95] is the stopping condition. Voronoi's algorithm stops as soon as all neighbouring perfect forms of a certain perfect form are isometric to some other perfect form already encountered. Algorithm 4.1 stops as soon as all perfect forms satisfying the upper bound in (4.46) have been enumerated.

The Fincke-Pohst algorithm is used to compute $\text{Min}(\boldsymbol{G})$ [78]. The toughest part of the algorithm is to compute the extreme rays of a polytope specified by linear inequalities. This is the bottleneck of enumerating non-isometric perfect

Table 4.1: An algorithm that traverses all perfect forms satisfying the bounds in (4.46).

Algorithm for Solving (4.32)

INPUT: A STARTING PERFECT FORM $\boldsymbol{G}_s$, E.G., THE ROOT LATTICE A_N.

OUTPUT: THE LIST OF $\boldsymbol{G}$ MATRICES CORRESPONDING TO THE VERTICES IN THE POLYTOPE $\mathcal{R}_1 \cap \{\boldsymbol{G} : \boldsymbol{G} \leq \text{ub}(\boldsymbol{S})\}$ THAT ARE CANDIDATES FOR SOLVING (4.36).

Let $\boldsymbol{G} = \boldsymbol{G}_s$ and define the boolean variable $b_{\boldsymbol{G}} \stackrel{\triangle}{=} 1$. Save the pair $(\boldsymbol{G}, b_{\boldsymbol{G}})$ in a set $\mathcal{G} = \{(\boldsymbol{G}, b_{\boldsymbol{G}})\}$ and $\boldsymbol{G}$ in a set $\mathcal{O} = \{\boldsymbol{G}\}$.

1. Compute $\text{Min}(\boldsymbol{G})$ and the extreme rays (edges) $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_k$ of the polyhedal cone

$$\{\boldsymbol{G}' \in \mathcal{S}^{N\times N} : \ \boldsymbol{G}'[\boldsymbol{x}] \geq 0 \ \forall \boldsymbol{x} \in \text{Min}(\boldsymbol{G})\}.$$

2. Determine neighbouring perfect forms $\boldsymbol{G}_i$ as $\boldsymbol{G}_i = \boldsymbol{G} + \alpha \boldsymbol{T}_i$, $i = 1 \ldots k$.

3. Let $\boldsymbol{G}_{j_1}, \ldots \boldsymbol{G}_{j_m}$ be those neighbouring forms satisfying the upper bound in (4.46) that also cannot be found in $\mathcal{G}$, and define $b_{\boldsymbol{G}_{j_l}} \stackrel{\triangle}{=} 0$, $l = 1 \ldots m$. Then let $\mathcal{G} = \mathcal{G} \cup \{(\boldsymbol{G}_{j_1}, b_{\boldsymbol{G}_{j_1}}), \ldots, (\boldsymbol{G}_{j_m}, b_{\boldsymbol{G}_{j_m}})\}$.

4. For each $\boldsymbol{G}_{j_l}$, $l = 1, \ldots, m$, check if there is a $\hat{\boldsymbol{G}} \in \mathcal{O}$ with eigenvalues majorizing the eigenvalues of $\boldsymbol{G}_{j_l}$. Then let $\mathcal{O} = \mathcal{O}/\{\hat{\boldsymbol{G}}\}$ and $\mathcal{O} = \mathcal{O} \cup \{\boldsymbol{G}_{j_l}\}$, i.e., exclude $\hat{\boldsymbol{G}}$ from $\mathcal{O}$ and include $\boldsymbol{G}_{j_l}$ into $\mathcal{O}$.

5. Find a pair $(\boldsymbol{G}_j, b_{\boldsymbol{G}_j})$ in $\mathcal{G}$ such that $b_{\boldsymbol{G}_j} = 0$. If such a pair exists, change the value of $b_{\boldsymbol{G}_j}$ to $b_{\boldsymbol{G}_j} = 1$, let $\boldsymbol{G} = \boldsymbol{G}_j$ and go to step 1. Otherwise, stop and return $\mathcal{O}$.

forms with Voronoi's algorithm and thereby solving the lattice packing problem in high dimensions. There exist methods that does this in $O(Nvd)$ time, where N is the dimension, d the number of non-redundant inequalities describing the polytope and v is the number of vertices in the polytope [97].

For step 2, determining the neighbouring perfect forms can be done by the algorithm in [84, Algorithm 2, Chapter 3], which computes the α needed in step 2. In the other steps, we use boolean variables $b_{\boldsymbol{G}_j}$ to denote whether a vertex has been visited or not.

A few comments regarding the complexity of solving (4.32) with Algorithm 4.1 compared to the method in Section 4.3.6. The latter method first uses Voronoi's algorithm to find non-isometric perfect forms, since this is less complex than enumerating Minkowski extreme lattices. The next step is to enumerate unimodular matrices satisfying the upper bound in (4.45). The former method works directly with Algorithm 4.1, which is based on enumerating perfect forms satisfying (4.46). Thus the question is: Is it more complex to directly enumerate vertices in $\mathcal{R}_1$ (Algorithm 4.1), or to only enumerate non-isometric vertices in $\mathcal{R}_1$, to then go over and enumerate unimodular matrices satisfying the bound in (4.45)? The answer to this complexity analysis is left for future work.

4.3.7 Suboptimal precoder construction

Finding the optimal solution is a computationally demanding task for today's computers, and suboptimal solutions are of interest. We base our suboptimal construction on $\boldsymbol{F}_{\text{gmd}}$ from Section 4.3.5.

It can be numerically verified that for $\boldsymbol{F}_{\text{gmd}}$, the received lattice is not a Minkowski extreme lattice, and is thereby not optimal. Hence, the performance of $\boldsymbol{F}_{\text{gmd}}$ can be improved by applying the result of Theorem 13. Let $\boldsymbol{B}_{\text{gmd}} = \boldsymbol{S}\boldsymbol{F}_{\text{gmd}}$ be the received lattice at the receiver, where $\boldsymbol{F}_{\text{gmd}}$ is scaled so that $D^2_{\min}(\boldsymbol{B}_{\text{gmd}}) = 1$. Now perform a Minkowski reduction on $\boldsymbol{B}_{\text{gmd}}$ by using the Minkowski reduction algorithm [86], so that we can factor the basis matrix as $\boldsymbol{B}_{\text{gmd}} = \boldsymbol{U}_{\text{gmd}}\boldsymbol{L}_{\text{gmd}}\boldsymbol{Z}_{\text{gmd}}$ for some rotation $\boldsymbol{U}_{\text{gmd}}$, Minkowski reduced lattice basis $\boldsymbol{L}_{\text{gmd}}$ with $D^2_{\min}(\boldsymbol{L}_{\text{gmd}}) = 1$, and unimodular $\boldsymbol{Z}_{\text{gmd}}$. It then follows that $\boldsymbol{F}_{\text{gmd}} = \boldsymbol{S}^{-1}\boldsymbol{U}_{\text{gmd}}\boldsymbol{L}_{\text{gmd}}\boldsymbol{Z}_{\text{gmd}}$. Define $\boldsymbol{F}_{m,i} \triangleq \boldsymbol{S}^{-1}\boldsymbol{U}_i\boldsymbol{L}_{m,i}\boldsymbol{Z}_{\text{gmd}}$, $i = 1 \ldots K$, to be the K different precoders where $\boldsymbol{L}_{m,i}$ is the i:th Minkowski extreme lattice with $D^2_{\min}(\boldsymbol{L}_{m,i}) = 1$, and $\boldsymbol{U}_i$ is the left orthogonal matrix of $\boldsymbol{L}_{m,i}\boldsymbol{Z}_{\text{gmd}}$. Applying Theorem 13, we know that $\text{tr}(\boldsymbol{F}^{\text{T}}_{m,j}\boldsymbol{F}_{m,j}) < \text{tr}(\boldsymbol{F}^{\text{T}}_{\text{gmd}}\boldsymbol{F}_{\text{gmd}})$ for some $1 \leq j \leq K$. Thus, $\boldsymbol{F}_{m,j}$ is a precoder performing better than the geometric mean precoder.

Hence, by performing a Minkowski reduction and using the resulting uni-

modular matrix together with one of the Minkowski extreme lattices, it is possible to improve upon the geometric mean precoder and reach closer to the lower bound given in (4.41). However, performing a Minkowski reduction includes finding the shortest basis vector in the lattice, which is an NP-hard problem [98]. Nevertheless, it turns out to be easily doable with a standard workstation at least for $N \lesssim 15$. Another method that can be used for this purpose is the iterative algorithm presented in [79].

4.3.8 Packing lattices are not always a solution to (4.32)

By applying the knowledge that perfect forms solve (4.32), we provide in this section a numerical example where the densest lattice packing is not a solution to (4.32). This shows that the solution to (4.30) is somewhat counter-intuitive: The optimal packing of points at the receiver, does not always minimize the total energy of the lattice points at the transmitter.

Note that a perfect form $\boldsymbol{G}$ that is a candidate for solving (4.32) must satisfy the upper bound (4.43). Since $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{e} \geq 1$, $\forall \boldsymbol{e} \in \mathbb{Z}^N/\{\boldsymbol{0}_N\}$, it holds that $g_{i,i} \geq 1$, $i = 1 \ldots N$, and thus $\mathrm{tr}(\boldsymbol{G}) \geq N$. Let λ_i denote the i:th shortest vector in the lattice $\boldsymbol{L}$ with Gram matrix $\boldsymbol{G}$. By definition, the minimum distance is $\lambda_1 = 1$. Now assume a channel $\boldsymbol{S}$ such that the upper bound in (4.43) is smaller than $(N-1)\lambda_1 + \lambda_2$. If $\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{Z}$ is another perfect form isometric to $\boldsymbol{G}$ that is also a candidate for solving (4.32), then $\mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{Z}) \leq (N-1)\lambda_1 + \lambda_2$. However, this inequality significantly limits the number of possible $\boldsymbol{Z}$ matrices and thus the number of perfect forms isometric to $\boldsymbol{G}$. Namely, since $\mathrm{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{Z}) = \sum_{j=1}^{N} \boldsymbol{Z}_j^{\mathrm{T}}\boldsymbol{G}\boldsymbol{Z}_j$ and each term $\lambda_1 \leq \boldsymbol{Z}_j^{\mathrm{T}}\boldsymbol{G}\boldsymbol{Z}_j \leq \lambda_2$, it follows that each $\boldsymbol{Z}_j$ must correspond to a minimum vector of $\boldsymbol{G}$, i.e., $\boldsymbol{Z}_j$ belongs to the set $\mathrm{Min}(\boldsymbol{G})$.

Let us apply this idea to 4-dimensional lattices. In 4 dimensions, there are only two non-isometric perfect forms, the D_4 and A_4 lattice. A Gram matrix for D_4[5] is

$$\boldsymbol{G}_{D_4} = \begin{pmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & -0.5 & 0 \\ 0.5 & -0.5 & 1 & -0.5 \\ 0 & 0 & -0.5 & 1 \end{pmatrix} \tag{4.49}$$

[5]Gram matrices for non-isometric perfect forms can be found at [99].

and for A_4,

$$\boldsymbol{G}_{A_4} = \begin{pmatrix} 1 & -0.5 & 0 & 0 \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & -0.5 \\ 0 & 0 & -0.5 & 1 \end{pmatrix}. \tag{4.50}$$

Hence, any perfect form in 4 dimensions can be expressed as either $\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{D_4}\boldsymbol{Z}$ or $\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{A_4}\boldsymbol{Z}$ for some unimodular $\boldsymbol{Z}$. Further, it holds that $\lambda_1 = 1$ and $\lambda_2 = 2$ for both $\boldsymbol{G}_{A_4}$ and $\boldsymbol{G}_{D_4}$. Now let $\boldsymbol{S}$ be

$$\boldsymbol{S} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.95 & 0 & 0 \\ 0 & 0 & 0.94 & 0 \\ 0 & 0 & 0 & 0.93 \end{pmatrix}. \tag{4.51}$$

The upper bound in (4.43) is 4.83 for this $\boldsymbol{S}$. Hence, if a perfect form $\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{D_4}\boldsymbol{Z}$ isometric to $\boldsymbol{G}_{D_4}$ solves (4.32), then the columns of $\boldsymbol{Z}$ must be taken from $\mathrm{Min}(\boldsymbol{G}_{D_4})$. Similarly, if a perfect form isometric to $\boldsymbol{G}_{A_4}$ solves (4.32), then the columns of the corresponding unimodular $\boldsymbol{Z}$ are taken from $\mathrm{Min}(\boldsymbol{G}_{A_4})$. It is an easy task to find $\mathrm{Min}(\boldsymbol{G}_{D_4})$ and $\mathrm{Min}(\boldsymbol{G}_{A_4})$, by applying the Fincke-Pohst algorithm, and also to find all unimodular matrices whose columns consist of these minimum vectors. Going through each perfect form obtained from these unimodular matrices, and plugging in the optimal precoder (4.37) into (4.32), the result is that the perfect form isometric to $\boldsymbol{G}_{A_4}$ gives the smallest value of the objective function in (4.32). Repeating the same argument for the channel

$$\boldsymbol{S} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.99 & 0 & 0 \\ 0 & 0 & 0.94 & 0 \\ 0 & 0 & 0 & 0.93 \end{pmatrix}, \tag{4.52}$$

one concludes that the perfect form isometric to $\boldsymbol{G}_{D_4}$ solves (4.32). Hence, this shows that both A_4 and D_4 occur as optimal lattice structures at the receiver; which one it is, depends on the channel $\boldsymbol{S}$.

4.3.9 Optimal lattices for low complexity ML decoders

Section 3.2 proposed suboptimal precoder constructions by also taking the ML decoding complexity into account. Clearly, the more non-zero elements in $\boldsymbol{G}$, the larger the complexity of the ML decoder. Assume that the ML decoder should have memory $M < N$. This forces $\boldsymbol{G}$ to have $N - M - 1$ diagonals that are zero, i.e., $\boldsymbol{G}$ is a banded symmetric matrix. Hence, it is of interest to solve

(4.2) over $\boldsymbol{G}$s that also satisfy this constraint, in addition to belonging to $\mathcal{R}_\lambda$. The framework developed in Section 4.3 enables one to do that. Constraining some elements $g_{j,k}$ in $\boldsymbol{G}$ to 0 simply corresponds to intersecting $\mathcal{R}_\lambda$ with the coordinate planes $g_{j,k} = 0$. This will give rise to a new polytope with new vertices, which correspond to new optimal lattices. Note that the constraint $g_{j,k} = 0$ means that the lattice basis vectors $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N]$ that represent $\boldsymbol{G}$, i.e., $\boldsymbol{G} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$, are such that $\boldsymbol{b}_j$ and $\boldsymbol{b}_k$ are orthogonal. Thus, forcing more elements $g_{j,k}$ to be zero results in more orthogonal optimal lattices. In [100], lattice precoding that produces a memory-1 ML decoder was proposed, which exhibits good BER performance as well for larger alphabets. Future research should investigate which optimal lattices that occur for different constraints on the ML decoding complexity.

4.4 Precoding with Complex-Valued Alphabets

As mentioned in the beginning of Section 4.3, precoding over the real-valued domain is more general than precoding over the complex-valued domain. Nevertheless, the results obtained in Section 4.3 are also applicable to complex-valued vectors and matrices. This is evident from the analysis in this section, where one can as well define a complex-valued Ryshkov polytope and thus perform the same analysis as before. We now investigate how this "complex-valued" Ryshkov polytope relates to the real-valued one studies in this section. Any $B \times B$ complex-valued Gram matrix $\boldsymbol{G}$ and any $B \times 1$ complex-valued error vector $\boldsymbol{e}$ can be extended to their real-valued equivalents $\boldsymbol{G}_r$ and $\boldsymbol{e}_r$ by means of the transformations in (3.22) and (3.23), respectively. The degrees of freedom in this skew-symmetric, real-valued matrix $\boldsymbol{G}_r$ is $B(B+1)$. Hence, we can define a $B(B+1)$ dimensional polytope

$$\mathcal{R}_\lambda^c \triangleq \{\boldsymbol{G}_r \ : \ \boldsymbol{e}_r^{\mathrm{T}}\boldsymbol{G}_r\boldsymbol{e}_r \geq \lambda\} \tag{4.53}$$

in the space of $B(2B+1)$ dimensional matrices. This polytope is an extension of the complex-valued Ryshkov polytope to the real-valued space. In this section, it was realized that the skew-symmetric constraint on $\boldsymbol{G}_r$ can be dropped, which made $\boldsymbol{G}_r$ full dimensional and $\mathcal{R}_c$ becomes equal to the Ryshkov polytope in Definition 7. However, if the skew-symmetric constraint is kept in order to deal with pure B dimensional complex-valued matrices, $\mathcal{R}_\lambda^c$ is only $B(B+1)$ dimensional. Furthermore, from the definition of $\mathcal{R}_\lambda^c$, it follows that any $\boldsymbol{G} \in \mathcal{R}_\lambda^c$ must be positive definite, and thus corresponds to a lattice with a minimum distance of at least 1. Thus, $\mathcal{R}_\lambda^c \subset \mathcal{R}$. This suggests that there can be vertices in $\mathcal{R}_\lambda$ that are not vertices in $\mathcal{R}_\lambda^c$, and explains also why there is room for

improvement by precoding over the real-valued domain instead of the complex-valued one.

Next, we argue that the results in Section 4.3 only cover the real-valued result in Theorem 8 in Section 4.2. Theorem 9 states that the optimal lattice for minimum distance in two dimensional complex-valued systems is unique, and corresponds to the Schläfli lattice in the four dimensional real-valued space. However, in Section 4.3.8, we proved that for some channels $\boldsymbol{S}$, the A_4 lattice occurs as an optimal point in $\mathcal{R}_1$. Thus, the general results in Section 4.3 only cover the real-valued precoding result in Section (4.2), but not the result in Theorem 9. Note that the channel $\boldsymbol{S}$ in Section 4.3.8 for which the A_4 lattice was optimal does not come from an extension of the form in (3.22), i.e., it can not represent the singular values of a complex-valued channel in two dimensions. Such $\boldsymbol{S}$ matrices must have pairs of eigenvalues on their diagonal. It is of future research to investigate which lattices that solve the complex-valued lattice problem.

4.5 Conclusions

This chapter studies precoding over non-singular linear channels with full CSI through a lattice-theoretic approach. The classical complex-valued linear channel is first transformed to a more general real-valued model which enables performance improvements over the classical complex-valued model. Then, the main problem studied in the work is to find lattices that maximize the minimum distance between the received lattice points, under an average energy constraint at the transmitter. The optimal lattice is analytically shown to be a perfect lattice, as defined by Ryshkov, for *any* given non-singular channel. Bounds on the optimal performance are developed, tighter than previously reported, which enable construction of algorithms that produce a finite codebook of matrices, from which the optimal precoder can be derived. Furthermore, a suboptimal precoder construction is presented together with bounds on its performance, which is analytically shown to improve upon a previous presented precoding scheme in the literature, by utilizing the new results in this work. In addition to this, we demonstrate with an example that optimal packing lattices are not always optimal for maximizing minimum distance, which is a counter-intuitive result at first sight. An immediate practical application of the derived results is precoding over large alphabets.

Chapter 5

Applications to Finite Alphabets

In Chapter 4, the theoretical analysis assumed an infinite signaling alphabet, which gives rise to an infinite, discrete error alphabet $\mathcal{E}^B$. Clearly, the infinite alphabet analysis becomes more valid the larger the signaling alphabet; however in practical systems, the alphabets can be rather small (e.g. binary and 4PAM real-valued models). Nevertheless, we saw in Sections 3.5.4 and 4.2.3 that the obtained lattice precoders occur as optimal solutions for the finite constellations as well. We will soon explain why this happens. Recall again that we can as well work with the real-valued model in (4.29), since any complex-valued communcation model can be transformed into a real-valued one. Moreover, as was argued in Section 4.3, precoding in the real-valued domain provides more degrees of freedom and thus better performance can be expected. Thus, the original minimum distance optimization problem in (3.6) is now assumed to be real-valued.

This chapter will present interesting observations on how the minimum distance problem behaves for finite alphabets. First, the chapter discusses the main difference between the problem studied in Chapter 4 and the finite alphabet case, and gives some interesting observations along with introducing new notation. Section 5.1 presents an efficient method to find the optimal solution to (3.6), given a certain $\boldsymbol{S}$, for small dimensions and alphabets. The observations that are made in that section are in agreement with the results of Section 3.5.2. Section 5.2 then presents new observations on the behavior of (3.6), which are then used as design guidelines in Section 5.4 to construct a finite codebook of precoders with excellent minimum distances. Section 5.3

presents a theorem that gives support to the observations made in Section 5.2 and the heuristic guideline presented therein. Finally, Section 5.5 presents simulation results with the new precoder codebook.

The main difference between assuming an infinite error alphabet compared to a finite one, is that the Gram matrix $\boldsymbol{G}$ does not have to be of full rank anymore, i.e., $\boldsymbol{G}$ can be a *degenerate* form, i.e., is of lower rank. Namely, since there are only finitely many error vectors $\boldsymbol{e}_j$, $j = 1, ..., m$, there are matrices $\boldsymbol{G}$ that satisfy $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{e} \geq 1$, but are not full rank. This implies that the optimal $\boldsymbol{G}$ is not necessarily a lattice in N dimensions, since a lattice in N dimensions has a full rank Gram matrix, and vice versa. Let $\boldsymbol{G} \succ 0$ denote that the Gram matrix $\boldsymbol{G}$ is positive semidefinite. Given a finite set $\mathcal{E}^N$ of N-dimensional integer (error) vectors, not containing the all-zero vector, we can define the "finite Ryshkov polytope" as

Definition 10. *The finite Ryshkov polytope is the set*

$$\mathcal{R}_\lambda(\mathcal{E}) = \{\boldsymbol{G} \ : \ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{e} \geq \lambda, \ \boldsymbol{e} \in \mathcal{E}^N, \ \boldsymbol{G} \succ 0\}$$

of all positive semidefinite forms (Gram matrices) $\boldsymbol{G}$ *with a minimum distance of at least* λ *over the set* $\mathcal{E}^N$*.*

For notational convenience, we omit the dimension N of the alphabet $\mathcal{E}^N$ in the definition of $\mathcal{R}_\lambda(\mathcal{E})$. Note that $\mathcal{R}_\lambda(\mathcal{E})$ is simply the region of all positive semidefinite $\boldsymbol{G}$ satisfying the constraints in (3.6), where the alphabet $\mathcal{E}^N$ is now finite. Furthermore, the shape of $\mathcal{R}_\lambda(\mathcal{E})$ clearly depends on the error vectors in the set $\mathcal{E}^N$. Figure 5.1 shows a simplified geometrical representation of $\mathcal{R}_\lambda(\mathcal{E})$. As with Figure 4.8, the boundary of the positive semidefinite cone corresponds to a complicated surface in higher dimensions, and is described by the solutions to a multidimensional polynomial equation [115]. Hence, it is a continuous curve, but its shape is complicated. This boundary consists of all positive semidefinite forms of rank $p < N$. Positive semidefinite forms of rank p define a surface of dimension $Np - p(p-1)/2$ in the $N(N+1)/2$ dimensional space of positive semidefinite forms.

From the definition of $\mathcal{R}_\lambda(\mathcal{E})$, it follows that $\mathcal{R}_\lambda \subset R_\lambda(\mathcal{E})$. More stringently, the following relation holds for $\mathcal{R}_\lambda(\mathcal{E})$. Let $\mathcal{E}_1, \mathcal{E}_2$ be two different alphabets where $\mathcal{E}_1 \subset \mathcal{E}_2$. Then it holds that

$$\mathcal{R}_\lambda(\mathcal{E}_2) \subseteq \mathcal{R}_\lambda(\mathcal{E}_1). \tag{5.1}$$

We can now reformulate (3.6) in terms of $\mathcal{R}_1(\mathcal{E})$. Herein, we let the dimension of (3.6) be N instead of B. Let $\mathcal{E}^N$ be some finite error alphabet. Factorize $\boldsymbol{F}$ in (3.6) as in (4.1). Then, it is clear that (3.6) is equivalent to the

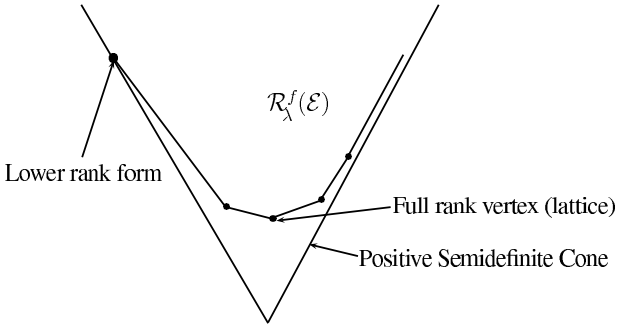

Figure 5.1: A visualization of $\mathcal{R}_\lambda(\mathcal{E})$ in the positive semidefinte cone. Note that in contrast to $\mathcal{R}_\lambda$, there are Gram matrices $\boldsymbol{G}$ in $\mathcal{R}_\lambda(\mathcal{E})$ that actually touch the positive semidefinite cone, and thus are degenerate positive semidefinite forms.

following optimization

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} = \arg\min_{\boldsymbol{W}} & \operatorname{tr}(\boldsymbol{B}^{\text{T}}\boldsymbol{W}^{\text{T}}\boldsymbol{S}^{-2}\boldsymbol{W}\boldsymbol{B}) \\ & \text{subject to} \\ & \boldsymbol{G} \in \mathcal{R}_1(\mathcal{E}) \\ & \boldsymbol{G} = \boldsymbol{B}^{\text{T}}\boldsymbol{B} \succ 0. \end{aligned} \tag{5.2}$$

Furthermore, Theorem 13 shows that the objective function in (5.2) is equivalent to $\min_{\boldsymbol{U}} \operatorname{tr}(\boldsymbol{U}\boldsymbol{S}^{-2}\boldsymbol{U}^{\text{T}}\boldsymbol{G})$, where $\boldsymbol{U}$ is an orthogonal matrix. Thus, (5.2) is also equivalent to

$$\begin{aligned} \boldsymbol{F}_{\text{opt}} = \arg\min_{\boldsymbol{U}} & \operatorname{tr}(\boldsymbol{U}\boldsymbol{S}^{-2}\boldsymbol{U}^{\text{T}}\boldsymbol{G}) \\ & \text{subject to} \\ & \boldsymbol{G} \in \mathcal{R}_1(\mathcal{E}) \end{aligned} \tag{5.3}$$

The relation in (5.1) implies that for finite alphabets $\mathcal{E}$, solving (5.3) by changing the constraint $\boldsymbol{G} \in \mathcal{R}_1(\mathcal{E})$ to $\boldsymbol{G} \in \mathcal{R}_1$, i.e., solving (5.3) over the Ryshkov polytope $\mathcal{R}_1$, can result in a non-optimal $\boldsymbol{G}$ to (5.3). The solution to (5.3), for a finite alphabet $\mathcal{E}$, could sometimes actually be attained at the boundary of the positive semidefinite cone, and soon we will show that this can indeed happen. Formulating (5.3) as in (4.36), a lower rank solution corresponds to a precoder that "turns off" the weak eigenmodes in $\boldsymbol{S}$, i.e., the N streams are multiplexed across the strong eigenmodes only.

Although $\mathcal{R}_1(\mathcal{E})$ and $\mathcal{R}_1$ are clearly different, the following interesting observations can be made about the solution to (5.3): If the optimal $\boldsymbol{G}$ to (5.3) is of full rank for some given $\boldsymbol{S}$ (such $\boldsymbol{S}$ exist, as will be discussed shortly) and some finite alphabet $\mathcal{E}$, then it must be one of the full rank vertices of $\mathcal{R}_1(\mathcal{E})$. The reason for this is Theorem 13, which shows that the objective function in (5.3) is concave over the set of positive semidefinite matrices, and thus also over $\mathcal{R}_1(\mathcal{E})$. As will be discussed shortly, there are matrices $\boldsymbol{S}$ for which the optimum to (5.3) is full rank. However, as for $\mathcal{R}_1^c$ in Section 4.4, it could happen that the full rank vertices of $\mathcal{R}_1(\mathcal{E})$ are not necessarily perfect lattices anymore, i.e., there are full rank vertices in $\mathcal{R}_1(\mathcal{E})$ that are not vertices of $\mathcal{R}_\lambda$ for any $\lambda > 0$. Nevertheless, as the alphabet $\mathcal{E}$ grows in size, there will be many common full rank vertices between $\mathcal{R}_1(\mathcal{E})$ and $\mathcal{R}_1$, and in the limit we know that all full rank $\boldsymbol{G}$ correspond to perfect lattices.

Further interesting observations can be noted from the relations in (5.1). Assume that we have the error alphabets $\mathcal{E}_{\text{Bin}}$ and $\mathcal{E}_{\text{4PAM}}$ resulting from the binary and 4PAM constellation, respectively. Then clearly, $\mathcal{E}_{\text{Bin}} \subset \mathcal{E}_{\text{4PAM}}$[1], and from (5.1), $\mathcal{R}_1(\mathcal{E}_{\text{4PAM}}) \subseteq \mathcal{R}_1(\mathcal{E}_{\text{Bin}})$. Assume an $\boldsymbol{S}$, for which the optimal $\boldsymbol{G}_{\text{Bin}}$ to (5.3) with $\mathcal{E} = \mathcal{E}_{\text{Bin}}$ is a full rank vertex of $\mathcal{R}_1(\mathcal{E}_{\text{Bin}})$. If $\boldsymbol{G}_{\text{Bin}}$ is also a vertex of $\mathcal{R}_1(\mathcal{E}_{\text{4PAM}})$, then clearly $\boldsymbol{G}_{\text{Bin}}$ also solves (5.3) with $\mathcal{E} = \mathcal{E}_{\text{4PAM}}$ for this specific $\boldsymbol{S}$. Actually, if $\boldsymbol{G}_{\text{Bin}} \in \mathcal{R}_1$, i.e., it corresponds to a perfect lattice of minimum distance 1, then since $\mathcal{R}_1 \subset \mathcal{R}_1(\mathcal{E}) \subseteq \mathcal{R}_1(\mathcal{E}_{\text{Bin}})$, for *any* finite $\mathcal{E}$ such that $\mathcal{E}_{\text{Bin}} \subset \mathcal{E}$, then $\boldsymbol{G}_{\text{Bin}}$ is a solution to (5.3) for the error alphabet $\mathcal{E}$ as well. In other words, if the lattice corresponding to $\boldsymbol{G}_{\text{Bin}}$ has a minimum distance of 1, then it solves (5.3) for the given $\boldsymbol{S}$, for *any* finite alphabet $\mathcal{E}$ that includes the binary error alphabet $\mathcal{E}_{\text{Bin}}$. Clearly, this argument can be extended to any alphabets $\mathcal{E}_1$ and $\mathcal{E}_2$ such that $\mathcal{E}_1 \subset \mathcal{E}_2$; binary and 4PAM alphabets were used only as an illustration of the argument.

Hence, it is of interest to classify the full rank vertices of $\mathcal{R}_1(\mathcal{E})$ for a certain error alphabet $\mathcal{E}$. If it turns out that the full rank vertices in $\mathcal{R}_1(\mathcal{E})$ with a minimum distance of 1, which are then perfect lattices, also appear as full rank vertices in $\mathcal{R}_1(\mathcal{E}')$ for any $\mathcal{E}'$ such that $\mathcal{E} \subset \mathcal{E}'$, then they are also candidates for solving (5.3) with $\mathcal{E} = \mathcal{E}'$. Numerical investigations suggest that in small dimensions, this might be the case. The `lrs` software in [97] was used to enumerate full rank vertices in two, three and four dimensions for different error alphabets $\mathcal{E}$. The error alphabets $\mathcal{E}$ are of the form $\mathcal{E} = \{-M, \ldots, M\}$,

[1] Note that this relation is not true if power scaling is used for the 4PAM constellation in order to have unit energy on the transmitted streams, since then the 4PAM error alphabet is not composed of integers. Thus, more rigorously, we mean that there is a constant $k > 0$ such that $\mathcal{E}_{\text{Bin}} \subset k\mathcal{E}_{\text{4PAM}}$. However, it is clear that this does not change the analysis, and WLOG we can assume PAM constellations that are not scaled.

for some integer M^2. Namely, in two dimensions, we enumerated all the full rank vertices of $\mathcal{R}_1(\mathcal{E})$ for $M \leq 10$. It turns out that for each such M, all vertices have minimum distance of 1, i.e., the vertices are (or isometric to) perfect lattices since they are uniquely determined from their minimum vectors. Clearly, in this case the vertices are isometric to the hexagonal lattice, since that is the only perfect lattice in two dimensions. In three dimensions, the same was done for $M \leq 4$, and for each M, the full rank vertices were isometric to the A_3 lattice. In four dimensions, for $M = 1$, we observed that all full rank vertices were isometric to either A_4 or D_4. Hence, it seems that once the alphabet $\mathcal{E}$ is symmetric, as in the case of error alphabets, the minimum vectors of the full rank Gram matrices $\boldsymbol{G}$ that are uniquely defined by a set of $N(N+1)/2$ equations in $\mathcal{R}_1(\mathcal{E})$ are already contained in the alphabet $\mathcal{E}^N$. A more rigorous investigation of this is left for future work.

Another interesting fact was noted in this enumeration of vertices. In three and four dimensions, some of the vertices are not full rank. In three dimensions, some of the vertices are of rank 2, and in four dimensions, of rank 3. Thus, even though they are uniquely defined by a set of $N(N+1)/2$ equations in $\mathcal{R}_1(\mathcal{E})$, these vertices touch the boundary of the positive semidefinite cone (note that this does not happen for the Ryshkov polytope $\mathcal{R}_1$). Hence, for some $\boldsymbol{S}$, it may happen that these degenerate precoders give the optimal minimum distance. Therefore, the simulation results for the two dimensional lattice precoders in Section 4.2 can be improved, by also including these precoders that turn off weak eigenmodes. This will be elaborated in Section 5.5. By using the lattice identification techniques in Section 3.5.2, we could identify the degenerate vertices found in three and four dimensions as perfect lattices of minimum distance 1, but now in a lower dimension. This confirms the observations made in Section 3.3 for complex-valued matrices, that for some $\boldsymbol{S}$, the optimal precoder gives rise to a lattice in lower dimensions.

Thus, the following knowledge about (5.3) is currently at hand. Given a matrix $\boldsymbol{S}$ for which the optimal $\boldsymbol{G}$ in (5.3) is of full rank, this $\boldsymbol{G}$ corresponds to a vertex in $\mathcal{R}_\lambda(\mathcal{E})$. Furthermore, tests in three and four dimensions show that these vertices are perfect lattices. On the other hand, if the solution is not at a full rank vertex of $\mathcal{R}_\lambda(\mathcal{E})$, then it is degenerate and is located on the boundary of the cone of positive semidefinite matrices. In this case, the solution does not necessarily represent a lattice.

[2]Note that error alphabets $\mathcal{E}$ that represent practical PAM constellations such as binary, 4PAM, 16PAM, etc., have an odd M.

5.1 Relation to Semidefinite Programming

For a fixed orthogonal matrix $\boldsymbol{U}$, the objective function in (5.3) is linear in $\boldsymbol{G}$, and thus (5.3) is a semidefinite programming problem (SDP). There are efficient numerical methods to solve semidefinite programs [101]. Hence, for a fixed $\boldsymbol{U}$, it is possible to calculate the optimal $\boldsymbol{G}$ for a given $\boldsymbol{S}$. Solving (5.3) for many different $\boldsymbol{U}$, it is possible to obtain solutions that are very close to the optimum. For small enough dimensions, the orthogonal matrices can be parameterized (e.g. by Givens rotations [102]) and (5.3) solved for each realization of $\boldsymbol{U}$.

This has been done for $N = 3$ and $\mathcal{E}_{\mathrm{Bin}}$, with the following outcome. For channels $\boldsymbol{S}$ that give a full rank optimal $\boldsymbol{G}$, we already know from the above mentioned enumeration with `lrs` that this $\boldsymbol{G}$ corresponds to a perfect lattice. This was now also confirmed by solving (5.3) for the parameterized $\boldsymbol{U}$ for many channels $\boldsymbol{S}$ that give rise to an optimal full rank $\boldsymbol{G}$. It is noteworthy that the channels $\boldsymbol{S}$ for which a full rank $\boldsymbol{G}$ is optimal consist of good eigenmodes, i.e., $s_{3,3}$ is quite close to $s_{1,1}$ (it turns out that for $s_{1,1}/s_{3,3} \leq 5/2$, the optimal $\boldsymbol{G}$ is full rank). Once $s_{3,3}$ becomes small compared to $s_{1,1}$, the optimal $\boldsymbol{G}$ degenerates, i.e., is of lower rank. However, interestingly, for many such $\boldsymbol{S}$, the optimal lower rank $\boldsymbol{G}$ still represents a lattice. Namely, for many different channels $\boldsymbol{S}$ for which the optimal $\boldsymbol{G}$ is of rank 2, this $\boldsymbol{G}$ of rank 2 represents the hexagonal lattice of minimum distance 1. In the case of an $\boldsymbol{S}$ that gives rise to a rank 1 $\boldsymbol{G}$, this $\boldsymbol{G}$ of rank 1 can be written as $\boldsymbol{G} = \boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}$, where $\boldsymbol{b} = [1\ 2\ 4]$ - hence, it is a one dimensional lattice. However, for the following $\boldsymbol{S}$,

$$\boldsymbol{S} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5.4}$$

the optimal $\boldsymbol{G}$ is of rank 2 and equals

$$\boldsymbol{G} = \begin{pmatrix} 1 & 2.4564 & -1.9564 \\ 2.4564 & 6.3636 & -4.6382 \\ 1.9564 & -4.6382 & 3.9128 \end{pmatrix}. \tag{5.5}$$

It is easily verified with the technique in Section 3.5.2 that $\boldsymbol{G}$ in (5.5) does not correspond to any lattice in two dimensions. By varying the second diagonal element $s_{2,2}$ in the interval $1.8540 \leq s_{2,2} \leq 2.1354$, different rank 2 optimal $\boldsymbol{G}$s are obtained. Hence, this corresponds to the fact that the optimum is at the boundary of the positive semidefinite cone, and more specifically, on the $Np - p(p-1)/2 = 5$ dimensional curve ($N = 3$, $p = 2$) that represents $\boldsymbol{G}$s of rank 2. Since this boundary is a continuous curve, varying the elements

in $\boldsymbol{S}$ continuously also varies the optimal solution in a continuous way. Thus, for these $\boldsymbol{S}$, the optimal solution is not structured in the nice way as was the case in Chapter 4, and the optimal solution does not change in a discrete fashion as with full rank vertices, but rather continuously with the channel $\boldsymbol{S}$. However, it turns out that the lattice solutions are close to the non-lattice $\boldsymbol{G}$ solutions, since once $s_{2,2}$ is outside the interval $[1.8540,\ 2.1354]$, the optimal $\boldsymbol{G}$ immediately becomes a lattice. Thus, since a very small set of $\boldsymbol{S}$ matrices give non-lattice solutions, a restriction to only use lattice precoders will not result in a significant loss in minimum distance.

If the alphabet $\mathcal{E}_{\text{Bin}}$ is increased to $\mathcal{E}_{\text{4PAM}}$, then the optimum solution for the channel in (5.4), and also to those channels for which $s_{2,2} \in [1.8540,\ 2.1354]$, $s_{1,1} = 10$, $s_{3,3} = 1$, suddenly becomes a $\boldsymbol{G}$ of rank 2 that represents the hexagonal lattice. Thus, increasing the alphabet results in optimal solutions which are lattices for even larger sets of $\boldsymbol{S}$. Moreover, the former rank 2 solution for these channels when using $\mathcal{E}_{\text{Bin}}$ does not satisfy the minimum distance constraint for the alphabet $\mathcal{E}_{\text{4PAM}}$. Hence, increasing the alphabet "cuts out" many of the lower rank solutions in $\mathcal{R}_1(\mathcal{E}_{\text{Bin}})$, since additional hyperplanes $\boldsymbol{e}^{\text{T}}\boldsymbol{G}\boldsymbol{e} \geq 1$ are added to the set $\mathcal{R}_1(\mathcal{E}_{\text{Bin}})$. Thus, it is then rare that the optimal solution to a certain $\boldsymbol{S}$ is of lower rank; an $\boldsymbol{S}$ for which this happens must be very ill-conditioned the larger the alphabet $\mathcal{E}$ becomes. Chapter 4 shows that in the limit, when the alphabet size $|\mathcal{E}|$ goes to infinity, the optimal $\boldsymbol{G}$ is never degenerate for any $\boldsymbol{S}$. However, once such an $\boldsymbol{S}$ is encountered for a finite alphabet, it is also very likely that it corresponds to a lower rank lattice solution. In Section 5.2, we will give plausible arguments to this behavior.

5.2 Relation to Quadratically Constrained Quadratic Programming

This section will make some observations on the behaviour of the optimal solution described in Section 5.1. Furthermore, these observations are used as guidelines in Section 5.4 to construct a finite codebook of precoders with large minimum distances.

The problem in (5.3) can also be formulated as a quadratic program over $\boldsymbol{B}$, where $\boldsymbol{G} = \boldsymbol{B}^{\text{T}}\boldsymbol{B}$. Let $\boldsymbol{b} = [\boldsymbol{b}_1^{\text{T}}\ \ldots\ \boldsymbol{b}_N^{\text{T}}]^{\text{T}}$ be a vectorization of $\boldsymbol{B}$, with the columns of $\boldsymbol{B}$ stacked on top of each other. Upon defining $\boldsymbol{C} \stackrel{\triangle}{=} \boldsymbol{I}_{N\times N} \odot \boldsymbol{S}^{-2}$, $\boldsymbol{E}_j \stackrel{\triangle}{=} \boldsymbol{e}_j\boldsymbol{e}_j^{\text{T}}$, $j = 1, \ldots, |\mathcal{E}^N|$, and $\boldsymbol{A}_j \stackrel{\triangle}{=} \boldsymbol{I}_{N\times N} \odot \boldsymbol{E}_j$, where $\odot$ denotes the matrix

Kroenecker product, (5.3) becomes

$$\begin{aligned} &\min_{\boldsymbol{b}} \operatorname{tr}(\boldsymbol{b}^{\mathrm{T}} \boldsymbol{C} \boldsymbol{b}) \\ &\text{subject to} \\ &\boldsymbol{b}^{\mathrm{T}} \boldsymbol{A}_j \boldsymbol{b} \geq 1, \quad j = 1, \ldots, |\mathcal{E}^N|. \end{aligned} \tag{5.6}$$

Hence, (5.6) is a quadratically constrained quadratic programming problem. The constraints in (5.6) correspond to the region that lies outside the union of ellipsoids $\boldsymbol{b}^{\mathrm{T}} \boldsymbol{A}_j \boldsymbol{b} \leq 1$, $j = 1, \ldots, |\mathcal{E}^N|$. Note however, from the definition of $\boldsymbol{A}_j$, that these ellipsoids are N dimensional ellipsoids in an N^2-dimensional space. Hence, geometrically, the problem is to find the smallest N^2-dimensional ellipsoid with the semiaxis specified by $\boldsymbol{C}$, such that it is not contained in the interior of the constraint region. Since the constraint region is not convex, (5.6) is a non-convex problem. In the case of an infinite integer alphabet, the constraint region consists of the union of infinitely many ellipsoids, and we know that the solution is at an intersection of $N(N+1)/2$ ellipsoids, which specify a unique point, up to rotation. In other words, the optimal $\boldsymbol{b}_{\mathrm{opt}}$ gives rise to $N(N+1)/2$ equalities in the constraint region of (5.6), and there are infinitely many optimal solutions $\boldsymbol{b}$ that can be expressed as $\boldsymbol{b} = \boldsymbol{U}_K \boldsymbol{b}_{\mathrm{opt}}$, where $\boldsymbol{U}_K = \boldsymbol{I}_{N \times N} \odot \boldsymbol{U}$ and $\boldsymbol{U}$ is any N-dimensional orthogonal matrix.

If $\boldsymbol{G}$ is of rank 1, then $\boldsymbol{b} = [b_1 \ \ldots \ b_N]$ is a vector of N elements, and the problem in (5.6) reduces to the following quadratic program with linear constraints

$$\begin{aligned} &\min_{\boldsymbol{b}} \|\boldsymbol{b}\|^2 \\ &\text{subject to} \\ &\sum_{j=1}^{N} e_{q,j} b_j \geq 1, \quad q = 1, \ldots, |\mathcal{E}^N| \end{aligned} \tag{5.7}$$

This problem is efficiently solved by the quadratic programming software in Matlab and its `quadprog` function. It was found that for many different error alphabets $\mathcal{E}$ and dimensions N, the optimal $\boldsymbol{b}$ is an integer vector which represents a one dimensional lattice. However, if the rank is increased, then the constraint region is no longer linear, and the problem becomes harder to solve.

Expanding (5.6) in the elements of $\boldsymbol{b}$, (5.6) can also be written directly in

the matrix elements $b_{j,k}$ as

$$\begin{gathered}\min_{\{b_{j,k}\}} \sum_{j=1}^{N} \frac{1}{s_{j,j}^2} \sum_{k=1}^{N} b_{j,k}^2 \\ \text{subject to} \\ \sum_{j=1}^{N} \left(\sum_{k=1}^{N} e_{q,k} b_{j,k} \right)^2 \geq 1, \quad q = 1, \ldots, |\mathcal{E}^N|.\end{gathered} \tag{5.8}$$

By performing the variable substitution $x_{j,k} = b_{j,k}^2$, the objective function in (5.8) becomes linear in this new coordinate system. Note that there is an ambiguity in this substitution, since the $b_{j,k}$ are uniquely defined by $x_{j,k}$ up to sign. Hence, the optimal solution to (5.8) can be found by solving

$$\begin{gathered}\min_{\{x_{j,k}\}} \sum_{j=1}^{N} \frac{1}{s_{j,j}^2} \sum_{k=1}^{N} x_{j,k} \\ \text{subject to} \\ \sum_{j=1}^{N} \left(\sum_{k=1}^{N} e_{q,k} \pm \sqrt{x_{j,k}} \right)^2 \geq 1, \quad q = 1, \ldots, |\mathcal{E}^N| \\ x_{j,k} \geq 0.\end{gathered} \tag{5.9}$$

for any realization of $\pm$ as either $+$ or $-$, and choosing the best out of these solutions. Note that the objective function is now simple, while the constraint region is complicated and is described by an intersection of spaces that lie above quadratic surfaces. However, this formulation can give a more clear geometrical explanation, especially in two dimensions. Note that if the constraint region in (5.9) would be contained inside a polyhedron located in the quadrant $x_{j,k} \geq 0$, with a finite number of vertices that all intersect the constraint region in (5.9), then the optimal solution to (5.9) would always be at one these vertices. In this case, the constraint region would contain some "isolated" points, as depicted in the left region in Figure 5.2. However, if the constraint region cannot be contained inside a polyhedron with vertices located on the constraint region, then the optimum to (5.9) for a certain $\boldsymbol{S}$ can lie on a curve with nonisolated points, as shown in the right region in Figure 5.2. In that case, varying the channel $\boldsymbol{S}$ continuously, a whole continuum of optimal $\boldsymbol{G}$s is obtained. When

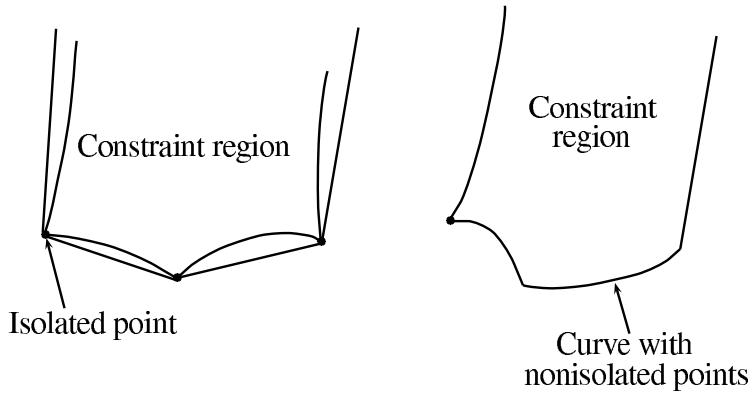

Figure 5.2: The isolated points are denoted as black dots. In the region to the left, the optimum to (5.9) always occurs at the isolated points. In the region to the right, there are points which are not isolated, so that the curve on which they are located is bent "outwards". Thus, minimizing a linear function over the region to the right gives a whole continuum of different solutions for different channels.

$N = 2$, finding the optimal $\boldsymbol{G}$ of rank 1 gives a simple problem in (5.9)

$$\begin{aligned} &\min_{\{x_1 \geq 0, x_2 \geq 0\}} x_1 + x_2 \\ &\text{subject to} \\ &(e_{q,1}\sqrt{x_1} \pm e_{q,2}\sqrt{x_2})^2 \geq 1, \quad q = 1, \ldots, |\mathcal{E}^N| \end{aligned} \tag{5.10}$$

For small alphabets $\mathcal{E}$, the two dimensional region in (5.10) is similar to the region to the left in Figure 5.2. Future analysis should investigate whether this is true for arbitrary dimensions N and alphabets $\mathcal{E}$.

However, from the numerical investigations in Section 5.1, we know that the constraint region in (5.9) does not look like the region on the left in Figure 5.2. Nevertheless, from the same numerical investigations, we know that there are portions of the constraint region in (5.9) that contain these isolated points. Note that these isolated points correspond to several constraints in (5.9) being active, and thus a heuristic approach to finding good solutions to (5.9) is to find those $\boldsymbol{G}$ in (5.3) that satisfy as many constraints as possible, even if they are of lower rank. Actually, the full rank solution satisfies as many equalities as possible, $N(N+1)/2$, which gives a set of isolated points to the constraint region. The lower rank optimal solutions obtained by the numerical

optimization in Section 5.1 also satisfy quite many equalities. Namely, for the three dimensional case of study, it turns out that the rank two solutions that correspond to two dimensional lattices satisfy 5 or more equalities (note that 5 is the dimension of the rank two matrix in three dimensions). However, the $\boldsymbol{G}$ in (5.5) only satisfies 4 equalities. This suggests that inducing more equalities on $\boldsymbol{G}$ for the symmetric alphabets $\mathcal{E}$ produces lattice solutions, even in the case of lower rank, and results in very good solutions to (5.3). It should however be emphasized that this is only a heuristic argument, and merely suggests a method that can construct a finite codebook of precoders with large minimum distances.

5.3 Least Number of Active Constraints in (5.3)

From Theorem 13, we can actually prove the following corollary regarding the least number of active constraints in (5.3) (i.e., the least number of equalities) that an optimal solution must have.

Corollary 4. *If the optimal solution $\boldsymbol{G}_o$ to (5.3) is of rank p, then at least $p(p+1)/2$ constraints are active in $\mathcal{R}_1(\mathcal{E})$.*

Proof. Factorize $\boldsymbol{G}$ in (5.3) as $\boldsymbol{G} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z}$, where $\boldsymbol{L}$ is a $p \times p$ matrix of rank p and $\boldsymbol{Z}$ is a $p \times N$ matrix. Note that $\boldsymbol{Z}$ is now a real-valued matrix, not to be confused with the unimodular matrix $\boldsymbol{Z}$ in (4.33). Hence, the optimization problem in (5.3) can now be written as

$$\begin{aligned} &\min_{\boldsymbol{U},\boldsymbol{L},\boldsymbol{Z}} \operatorname{tr}(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{S}^{-2}\boldsymbol{U}\boldsymbol{L}\boldsymbol{Z}) \\ &\qquad \text{subject to} \\ &\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z}\boldsymbol{e}_j \geq 1, \quad j = 1, \ldots, |\mathcal{E}^N|. \end{aligned} \tag{5.11}$$

Denote $\boldsymbol{L}_o$ and $\boldsymbol{Z}_o$ as the optimal solution to (5.11), i.e., $\boldsymbol{G}_o = \boldsymbol{Z}_o^{\mathrm{T}}\boldsymbol{L}_o^{\mathrm{T}}\boldsymbol{L}_o\boldsymbol{Z}_o$. Inserting $\boldsymbol{Z}_o$ into the objective function in (5.11), an optimization over $\boldsymbol{U}$ and $\boldsymbol{L}$ is left. Theorem 13 shows that for any fixed $\boldsymbol{Z}$, the objective function in (5.11) is concave over $\boldsymbol{G}_{\boldsymbol{L}} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$. Moreover, the constraint region in (5.11), for a fixed $\boldsymbol{Z}$, is a finite Ryshkov polytope in $p(p+1)/2$ dimensions, but where the alphabet is given by the vectors $\boldsymbol{Z}\boldsymbol{e}_j$, $j = 1, \ldots, |\mathcal{E}^N|$. Thus, the optimal $\boldsymbol{G}_{\boldsymbol{L}_o}$ is inside this Ryshkov polytope. If $\boldsymbol{G}_{\boldsymbol{L}_o}$ would be or rank lower than p, then $\boldsymbol{G}_o = \boldsymbol{Z}_o^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}_o}\boldsymbol{Z}_o$ is also or rank lower than p, contradicting the hypothesis of the theorem. Hence, this implies that the optimal $\boldsymbol{G}_{\boldsymbol{L}_o}$ must be a full rank vertex of this finite Ryshkov polytope, and thus satisfy at least $p(p+1)/2$ equalities in the constraint region of (5.11). Hence, this proves the corollary. □

Corollary 4 shows that the higher rank the optimal solution has, the more equalities it satisfies in the constraint region. Moreover, it provides a lower bound to the number of active constraints for the optimal solution. This supports the heuristic arguments in Section 5.2, which propose to look for solutions to (5.3) that satisfy as many equalities as possible.

5.4 Finite Codebook of Lattice Precoders

Adhering to the structure of the optimal solutions in three dimensions, presented in Section 5.1, and the heuristic arguments in Section 5.2, we are interested in finding lattice precoders of different ranks, which should be found by enforcing enough equalities in $\mathcal{R}_1(\mathcal{E})$. It is expected that the resulting precoders will have large minimum distances, and it might be that they are optimal for many $\boldsymbol{S}$ as was observed in Section 5.1; note, however, that the optimality cannot be guaranteed. As the results in Section 5.1 show, the lower rank $\boldsymbol{G}$s are also perfect lattices, as well as the full rank $\boldsymbol{G}$s. To enumerate the full rank $\boldsymbol{G}$s for a certain alphabet, we use the `lrs` software. However, to enumerate the lower rank lattices, we develop a novel method for that purpose.

Factorize a rank $p < N$ lattice Gram matrix $\boldsymbol{G}$ as $\boldsymbol{G} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{Z}$, with $\boldsymbol{L}$ a full rank $p \times p$ matrix and $\boldsymbol{Z}$ an integer $p \times N$ matrix. The generator matrix $\boldsymbol{L}$ is chosen as a generator matrix for a perfect lattice in p dimensions with a minimum distance of 1, hence $\boldsymbol{G}_{\boldsymbol{L}} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}$ is a Gram matrix for a perfect lattice. Next, we need to determine the integer matrices $\boldsymbol{Z}$ that give at least K equalities in $\mathcal{R}_1(\mathcal{E})$. Complying with the full rank case, where the optimal solution satisfies $N(N+1)/2$ equalities, we will set $K = Np - p(p-1)/2$, since that is the dimensionality of a rank p matrix. Note that $Np - p(p-1)/2 > p(p+1)/2$, hence this is a stronger requirement than the result in Corollary 4. Each constraint $\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}}\boldsymbol{Z}\boldsymbol{e}_j \geq 1$ can be written in a vectorized form

$$\boldsymbol{e}_j^{\mathrm{T}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}_{\boldsymbol{L}}\boldsymbol{Z}\boldsymbol{e}_j = \boldsymbol{z}^{\mathrm{T}}\boldsymbol{K}_j\boldsymbol{z} \geq 1, \quad j = 1, \ldots, |\mathcal{E}^N|, \tag{5.12}$$

where $\boldsymbol{z} = [\boldsymbol{z}_1^{\mathrm{T}} \ \ldots \ \boldsymbol{z}_N^{\mathrm{T}}]^{\mathrm{T}}$ is a vectorization of $\boldsymbol{Z}$, and $\boldsymbol{K}_j = \boldsymbol{E}_j \odot \boldsymbol{G}_{\boldsymbol{L}}$, with $\boldsymbol{E}_j = \boldsymbol{e}_j\boldsymbol{e}_j^{\mathrm{T}}$. Each $\boldsymbol{K}_j$ is an $Np \times Np$ matrix of rank p. Next, we factorize each $\boldsymbol{K}_j$ as $\boldsymbol{K}_j = \boldsymbol{R}_j^{\mathrm{T}}\boldsymbol{R}_j$, where $\boldsymbol{R}_j$ is an $Np \times Np$ upper triangular matrix of rank p. This can be done, by first finding the eigenvalue decomposition of $\boldsymbol{K}_j$, $\boldsymbol{K}_j = \boldsymbol{U}_j\boldsymbol{\Sigma}_j\boldsymbol{U}_j^{\mathrm{T}}$ and letting $\boldsymbol{T}_j = \sqrt{\boldsymbol{\Sigma}_j}\boldsymbol{U}_j^{\mathrm{T}}$. Next, we perform a QR factorization of $\boldsymbol{T}_j$, $\boldsymbol{T}_j = \boldsymbol{Q}_j\boldsymbol{R}_j$, for some orthogonal $\boldsymbol{Q}_j$ and upper triangular $\boldsymbol{R}_j$. Hence, $\boldsymbol{K}_j = \boldsymbol{T}_j^{\mathrm{T}}\boldsymbol{T}_j = \boldsymbol{R}_j^{\mathrm{T}}\boldsymbol{R}_j$. From this, each constraint in (5.12)

becomes

$$\boldsymbol{z}^{\mathrm{T}}\boldsymbol{R}_j^{\mathrm{T}}\boldsymbol{R}_j\boldsymbol{z} \geq 1. \tag{5.13}$$

If the kth constraint, $1 \leq k \leq |\mathcal{E}^N|$, is active, this corresponds to finding those integer vectors $\boldsymbol{z}$ for which $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{R}_k^{\mathrm{T}}\boldsymbol{R}_k\boldsymbol{z} = 1$. Note that this would be a simple sphere decoding algorithm if $\boldsymbol{R}_k$ was of full rank Np, however since every $\boldsymbol{R}_j$ is of rank p, the classical sphere decoding algorithm cannot be applied. Instead, we note the structure of each $\boldsymbol{R}_j$. Let $\mathrm{chol}(\boldsymbol{G})$ denote the Cholesky factorization of a positive definite matrix $\boldsymbol{G}$. From the definition of $\boldsymbol{R}_j$, it follows that each $\boldsymbol{R}_j$ is of the form

$$\boldsymbol{R}_j = \begin{pmatrix} e_{j,1}\mathrm{chol}(\boldsymbol{G}_{\boldsymbol{L}}) & e_{j,2}\mathrm{chol}(\boldsymbol{G}_{\boldsymbol{L}}) & \dots & e_{j,N}\mathrm{chol}(\boldsymbol{G}_{\boldsymbol{L}}) \\ \boldsymbol{0}_{Np-p,p} & \boldsymbol{0}_{Np-p,p} & \dots & \boldsymbol{0}_{Np-p,p} \end{pmatrix}. \tag{5.14}$$

Due to the symmetry of the error alphabet $\mathcal{E}$, there are error vectors $\boldsymbol{e}_j$ such that the elements $e_{j,l} \neq 0$ and $e_{j,k} = 0$ for $k \neq l$ (a coordinate vector). Let $\boldsymbol{e}_1$ be the first coordinate vector, i.e., $e_{1,1} \neq 0$, $e_{1,k} = 0$, $k > 1$. If it would hold that $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{z} \leq u_1$, where u_1 is a given upper bound, then we can find the p coordinates $z_{1:p}$ of $\boldsymbol{z}$ by running the sphere decoding algorithm with $\boldsymbol{R}_1$ and the upper bound u_1. For each such decoded vector $z_{1:p}$ of p coordinates, we let $\boldsymbol{z} = [z_{1:p}^{\mathrm{T}}\ \boldsymbol{0}_{1,Np-p}]^{\mathrm{T}}$ and put $\boldsymbol{z}$ in a list $\mathcal{Z}_0$. Next, an error vector $\boldsymbol{e}_2$ is chosen such that $e_{2,1} \neq 0$ and $e_{2,2} \neq 0$, but $e_{2,k} = 0$ for $k > 2$. Multiplying each vector $\boldsymbol{z}_j$ in $\mathcal{Z}_0$ with $\boldsymbol{R}_2$ results in a vector $\boldsymbol{v}_j$. Now, for each such $\boldsymbol{v}_j$, we can decode the coordinates $z_{p+1:2p}$ by running the sphere decoder with $\boldsymbol{R}_2$ and providing it the sphere center $-\boldsymbol{v}_j$, along with a new upper bound u_2. Each newly decoded vector of p coordinates is combined with $\boldsymbol{z}_j$ and put into a new list $\mathcal{Z}_1$ as $\mathcal{Z}_1 = \mathcal{Z}_1 \cup \{[z_{j,1:p}^{\mathrm{T}}\ z_{p+1:2p}^{\mathrm{T}}]^{\mathrm{T}}\ \boldsymbol{0}_{1,Np-2p}\}$. In the next step, we choose the vector $\boldsymbol{e}_3$ with the first three coordinates non-zero, and continue this process until all coordinates are decoded. At the end, a list $\mathcal{Z}_N$ is obtained with those $\boldsymbol{z}$ satisfying the upper bound u_j at each step j. Finally, those $\boldsymbol{z}$ that give less than K equalities in (5.13) are discarded. Hence, the main importance in this approach are the upper bounds $u_0, \dots, u_{N-1}$. They are clearly related to the minimum distance profile of the optimal solution, since they bound the distance of the optimal solution for the different error vectors. Finding good such upper bounds on the minimum distance are left for future research. The discussed algorithm is summarized by the pseudo code in Table 5.1.

Table 5.1: An algorithm that enumerates rank p perfect lattices of dimension N, that are candidates for solving (5.3).

Algorithm for Enumerating Lattice Precoders

INPUT: A p-DIMENSIONAL PERFECT FORM $\boldsymbol{G}$, THE DIFFERENT $\boldsymbol{R}_j$ IN (5.14) WITH $\boldsymbol{e}_j$ CONTAINING j NON-ZERO ELEMENTS, UPPER BOUNDS $u_1, \ldots, u_N$, AND THE NUMBER OF EQUALITIES K.

OUTPUT: THE LIST $\mathcal{G}$ OF $N \times N$ GRAM MATRICES CORRESPONDING TO THE PERFECT FORM $\boldsymbol{G}$ THAT GIVE AT LEAST K EQUALITIES IN $\mathcal{R}_1(\mathcal{E})$.

1. Initialize $j = 0$, $\mathcal{Z}_j = \{\}$ and $\mathcal{G} = \{\}$.

2. Set $\mathcal{Z}_{j+1} = \{\}$. For each $\boldsymbol{z} = [z_{1:jp}^{\mathrm{T}} \; \boldsymbol{0}_{1,Np-jp}]^{\mathrm{T}} \in \mathcal{Z}_j$, take $\boldsymbol{R}_{j+1}$ and construct $\boldsymbol{v}_{j+1} = \boldsymbol{R}_{j+1}\boldsymbol{z}$. Decode the p coordinates $z_{jp+1:(j+1)p}$ of $\boldsymbol{z}$ by applying the sphere decoder, operating with the sphere center $-\boldsymbol{v}_{j+1}$ and the upper bound u_{j+1}. For each newly decoded $z_{jp+1:(j+1)p}$, let $\mathcal{Z}_{j+1} = \mathcal{Z}_{j+1} \cup \{[z_{1:jp}^{\mathrm{T}} \; z_{jp+1:(j+1)p}^{\mathrm{T}} \; \boldsymbol{0}_{1,Np-2p}]^{\mathrm{T}}\}$. Repeat until all $\boldsymbol{z}$ in $\mathcal{Z}_j$ have been traversed.

3. If $j = N - 1$, go to step 4, otherwise set $j = j + 1$ and go to step 2.

4. Check if any of the vectors $\boldsymbol{z}$ in $\mathcal{Z}_N$ violate the upper bound $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{R}_j\boldsymbol{z} \leq u_N$ for some $1 \leq j \leq |\mathcal{E}^N|$, or give less than K equalities in the system $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{R}_j\boldsymbol{z} \geq 1$, $j = 1, \ldots, |\mathcal{E}^N|$. Exclude those $\boldsymbol{z}$ from the list $\mathcal{Z}_N$. For each $\boldsymbol{z}$ in the resulting $\mathcal{Z}_N$, construct the matrix $\boldsymbol{Z}$ and let $\mathcal{G} = \mathcal{G} \cup \{\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{Z}\}$. Exclude elements from $\mathcal{G}$ by the majorization principle described in Section 4.3.6. Return $\mathcal{G}$.

5.5 Numerical Results for Finite Alphabet Lattice Precoders

In this section, we present information rate and SER simulations for the lattice precoders that were found with the `lrs` software in [97], the lower rank lattice enumeration algorithm in Table 5.1 and the quadratic programming formulation for rank 1 in (5.7). Since (5.7) produces one global solution, the optimal rank 1 precoder for a certain alphabet $\mathcal{E}$ and N is easy to find and does not depend on the channel realization $\boldsymbol{S}$. We investigate the following scenarios

- Scenario 1: $N = 3$, real-valued parallel Gaussian channels, with an error alphabet $\mathcal{E}_{\text{Bin}}^N$ coming from a binary constellation.
- Scenario 2: $N = 3$, real-valued parallel Gaussian channels, with an error alphabet $\mathcal{E}_{\text{4PAM}}^N$ coming from a 4PAM constellation.
- Scenario 3: $N = 4$, real-valued parallel Gaussian channels, with an error alphabet $\mathcal{E}_{\text{Bin}}^N$ coming from a binary constellation.
- Scenario 4: $N = 4$, real-valued parallel Gaussian channels stemming from a complex-valued 2×2 Rayleigh fading MIMO model, with an error alphabet $\mathcal{E}_{\text{Bin}}^N$ coming from a binary constellation.

In all these scenarios, the noise is real-valued Gaussian noise with mean zero and variance 1/SNR over each stream. The binary constellation is $\mathcal{E}_{\text{Bin}}^N = \{\pm 1\}$, while $\mathcal{E}_{\text{4PAM}}^N = \{\pm 3/\sqrt{5}, \pm 1/\sqrt{5}\}$. Moreover, the SER is an average over all channel realizations. We now present the sizes of the codebooks obtained from `lrs` and the algorithm in Table 5.1 for the different scenarios.

- Scenario 1: The final codebook contains 8 lattice precoders. There are 3 of rank 3 and 4 of rank 2.
- Scenario 2: The final codebook contains 233 lattice precoders. There are 198 of rank 3 and 34 of rank 2.
- Scenario 3: The final codebook contains 50 lattice precoders. There are 26 of rank 4, 19 of rank 3 and 4 of rank 2.

Hence, the final codebooks are quite small for these dimensions and alphabets.

Figure 5.3 shows a SER simulation for Scenario 1. The codebook of 8 lattice precoders is compared with a random codebook of 8 precoders, a random codebook of 64 precoders, the geometric mean decomposition (GMD) precoder in [92] and no precoding. It is seen that the lattice precoder codebook improves significantly upon the competing schemes. Moreover, increasing the size of the

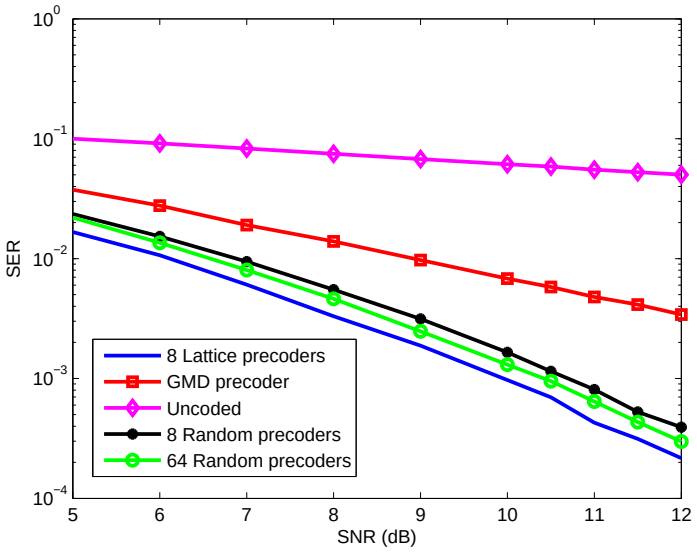

Figure 5.3: SER comparison for $N = 3$ with binary signaling. The comparison is made between the codebook of 8 lattice precoders, two random codebooks of 8 and 64 precoders, respectively, the GMD precoder in [92], and no precoding. There is a clear performance gain by using the lattice precoders.

random codebook from 8 to 64 precoders certainly improves its performance, however as seen in Figure 5.3, the performance is still a loss compared to the lattice precoder codebook.

Figure 5.4 shows a SER simulation for Scenario 3. The codebook of 233 lattice precoders again performs significantly better than a random codebook of the same size and the GMD precoder. The non-precoding case is not shown, since its loss is very big. In Figure 5.5, a SER simulation for Scenario 3 is shown. Similar conclusions can be derived as with the previous cases, where we notice that the gap between the lattice precoders and the other schemes is even wider now.

Next, we turn to a 2×2 complex-valued Rayleigh fading MIMO channel with 4QAM, as in Section 4.2.3. The two dimensional complex-valued model is now extended to 4 real-valued dimensions over which precoding is performed, which provides more degrees of freedom as argued in Section 4.3. This amounts to precoding over a BPSK alphabet in 4 dimensions, and thus we can use the

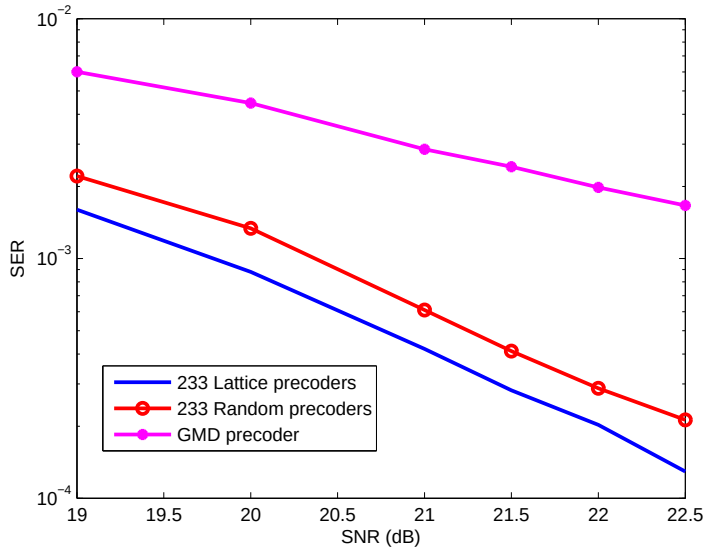

Figure 5.4: SER comparison for $N = 3$ with 4PAM signaling. The comparison is between the codebook of 233 lattice precoders, a random codebook of 233 precoders, the GMD precoder in [92], and no precoding. Again, there is a performance gain by using the lattice precoders.

lattice precoder codebook for this purpose; hence, this is Scenario 4 above. The SER simulation is shown in Figure 5.6. The lattice precoding codebook of 50 precoders is compared to the optimal two-dimensional complex-valued precoder in [67] and the GMD precoder. It is seen that the performance of the lattice codebook is the same as of the optimal complex-valued precoder, and the GMD precoder is once again far off in performance. Hence, in this case, the additional degrees of freedom provided in 4 dimensional real-valued space do not have a significant impact on the SER performance. However, this simulation shows that the lattice precoding results in Section 4.2.3 can be improved significantly, since now there are lattice precoders that avoid transmitting streams over weak eigenmodes, which in general lowers the minimum distance. Hence, the lattice precoders can perform as well as the optimum complex-valued precoder for 4QAM. More interestingly, even though the SER performance of the lattice precoders is the same as of the optimal complex-valued precoder, it turns out that for all the tested channels, the minimum distance of the lattice precoders is

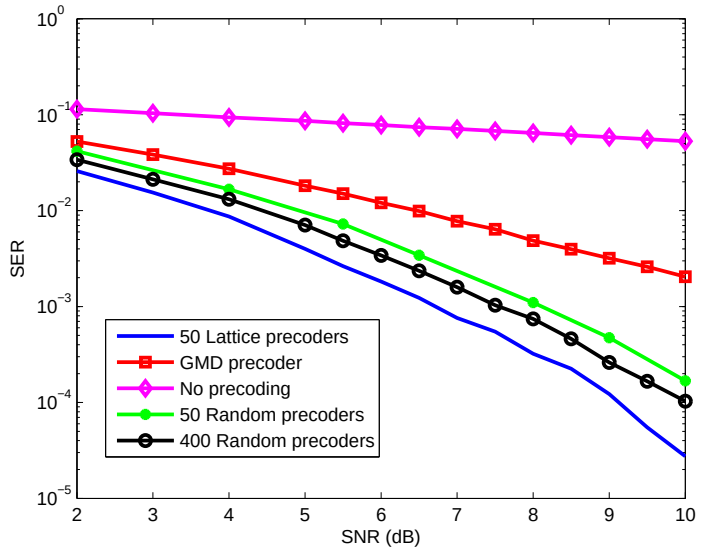

Figure 5.5: SER comparison for $N = 4$ and binary signaling. The comparison is between the codebook of 50 lattice precoders, a random codebook of 50 precoders and another one of 400 precoders, the GMD precoder in [92], and no precoding. Again, there is a clear performance gain by using the lattice precoders.

equal to or larger than the the minimum distance of the optimal complex-valued precoder. Namely, for those channels that result in an optimal full rank lattice precoder, the minimum distance of the lattice precoder and the complex valued precoder is the same. For channels that result in lattice precoders of lower rank, the minimum distance of the complex valued precoders is less. Thus, the optimal complex valued precoder produces a Gram matrix that is a full rank vertex in $\mathcal{R}_\lambda(\mathcal{E}_{\text{Bin}})$, and for the channels that arise in the simulation, only one vertex is optimal. However, as soon as they turn off transmision across weak eigenmodes, the lower rank lattice precoders give a higher minimum distance. On average, the minimum distance of the lattice precoders is 7% higher than the optimal complex-valued precoder, and thus this slight gain does not result in a significant SER gain. Nevertheless, this is of theoretical interest, since it shows that the obtained lattice precoders truly operate close to the optimum.

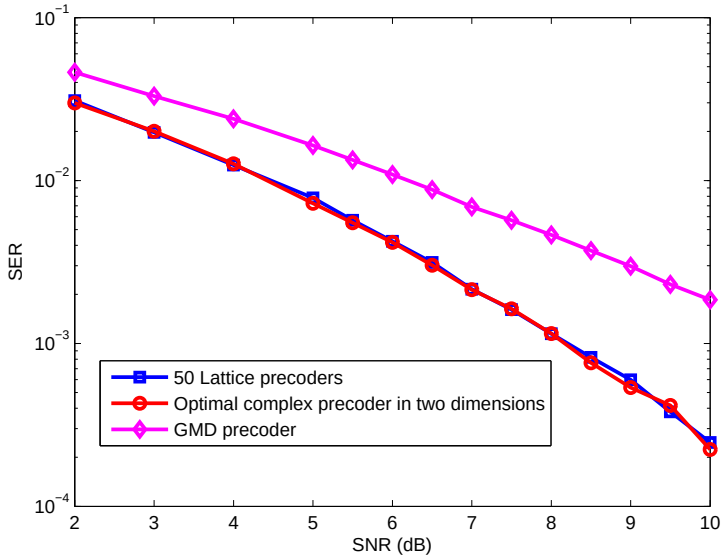

Figure 5.6: SER comparison for $N = 4$ and binary signaling, but for channels that arise from a 2×2 complex-valued MIMO model. The comparison is between the codebook of 50 lattice precoders, the optimal complex-valued precoder in two dimensions from [67] and the GMD precoder in [92]. Note that the 50 lattice precoders operate as well as the optimal complex-valued precoder, thus improving the performance in Section 4.2.3.

We move on to information rate curves for the obtained precoders. From the results in [83], we expect high information rates as the SNR gets larger, since precoders that maximize the minimum distance also maximize the information rate in the high SNR regime. Figures 5.7, 5.8 and 5.9 show ergodic information rates for Scenario 1, 2 and 3, respectively. The information rates of the lattice precoders are compared with 1) capacity, i.e., real-valued Gaussian symbols with waterfilling, 2) constrained capacity with Gaussian symbols and uniform power distribution, 3) a unitary precoder that equals the DFT matrix, which spreads the information bits evenly across the channel, 4) no precoding. As seen, in all cases, the information rate of the lattice precoder codebook is high, and comes quite close to the capacity for moderate and low SNRs as well. At higher SNRs, it converges quickly to the maximum limit compared with

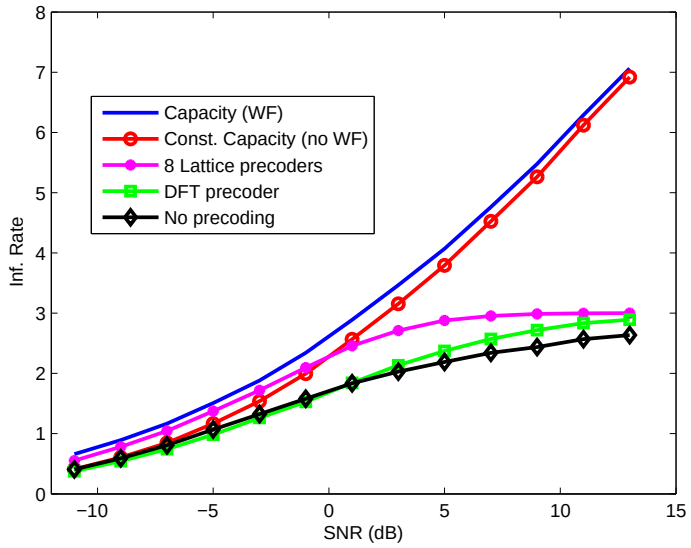

Figure 5.7: Ergodic information rates for $N = 3$ and binary signaling. The comparison is between the capacity, the constrained capacity, the 8 lattice precoders, a DFT precoder and no precoding. The information rate of the lattice precoders is close to capacity and converges quickly to the maximum of 3 bits per channel use.

the unitary precoders, and this behavior is expected as previously discussed. Thus, beside providing strong SER results, the obtained lattice precoders also produce high information rates, and this is a very desirable property.

In Figure 5.10, an interesting observation is made. It shows the information rate for $N = 4$ with binary signaling across a fixed channel $\boldsymbol{H}$, for different SNRs. For this particular $\boldsymbol{H}$, the information rate of the lattice precoders is extremely close to the capacity at low and moderate SNRs. This is an appealing behavior, which shows that even in the low SNR regime, there is a possibility to reach close to optimal rates across certain channels by using only a small set of precoders which also provide very good SER performance. Such channels also exist for $N = 3$. However, Figure 5.11 shows another channel, where the information rate of the lattice precoders is further away from the capacity than in Figure 5.10. The average performance is of course dictated by the ergodic capacity. Nevertheless, it is interesting that the performance of

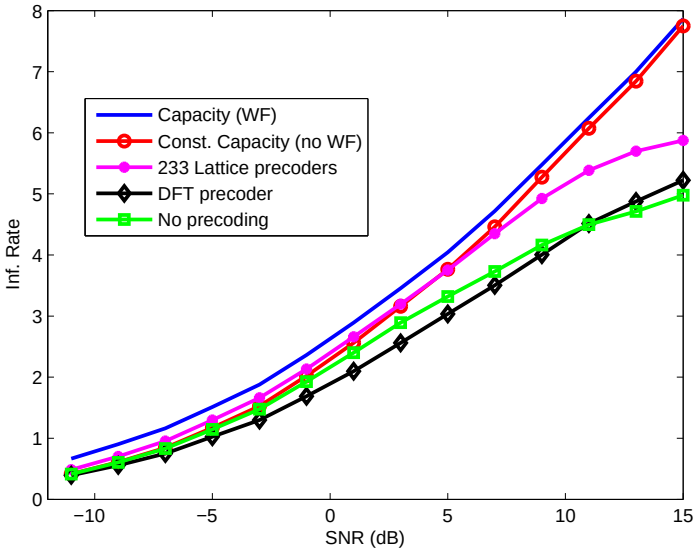

Figure 5.8: Ergodic information rates for $N = 3$ and 4PAM signaling. The comparison is between the capacity, the constrained capacity, the 233 lattice precoders, a DFT precoder and no precoding. The information rate of the lattice precoders is again quite close to capacity and converges quickly to the maximum of 6 bits per channel use.

precoders maximizing the minimum distance is also very good at low SNRs for certain channel outcomes.

As a closing remark, we note that in all simulations, it never happened that the lattice precoder codebook had inferior minimum distance to any of the other schemes presented herein. Thus, it is believed that these precoders truly operate close to the upper limit (in terms of minimum distance). Note also that the GMD precoder is far away in performance compared to the other schemes. The main reason for this is that the GMD precoder always transmits across all the channel eigenmodes, which thus can produce small distances in case of ill-conditioned channels. However, as shown in Section 4.3.7, the GMD precoder provides close to optimal minimum distances for large alphabets (i.e., for those cases when it is never favourable to turn off weak eigenmodes, since the alphabet is large). Hence, a method to improve the GMD is also to include

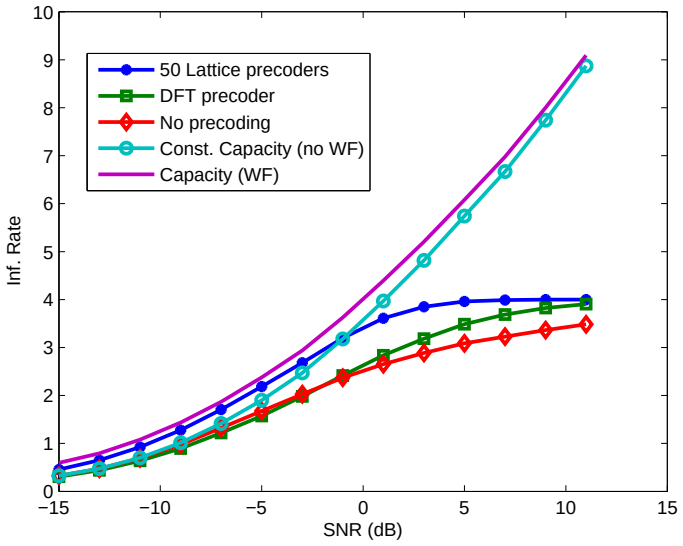

Figure 5.9: Ergodic information rates for $N = 4$ and binary signaling. The comparison is between the capacity, the constrained capacity, the 50 lattice precoders, a DFT precoder and no precoding. Once again, the information rate of the lattice precoders is close to capacity and converges quickly to the maximum of 4 bits per channel use.

bit loading, which would provide it with a well-conditioned channel and thus improve its performance.

5.6 Conclusions

This chapter shows that utilizing the lattice theoretic techniques developed in Chapter 4, it is possible to construct lattice precoders that provide excellent SER performance for finite alphabets as well. Heuristic arguments are presented that explain why this happens, and they suggest to look for Gram matrices of different rank that have many active constraints in the finite Ryshkov polytope. This property is satisfied by lattice precoders of degenerate rank. By studying the optimal solution in small dimensions, it is realized that lattice precoders are in many cases optimal and in other cases close to optimal. A

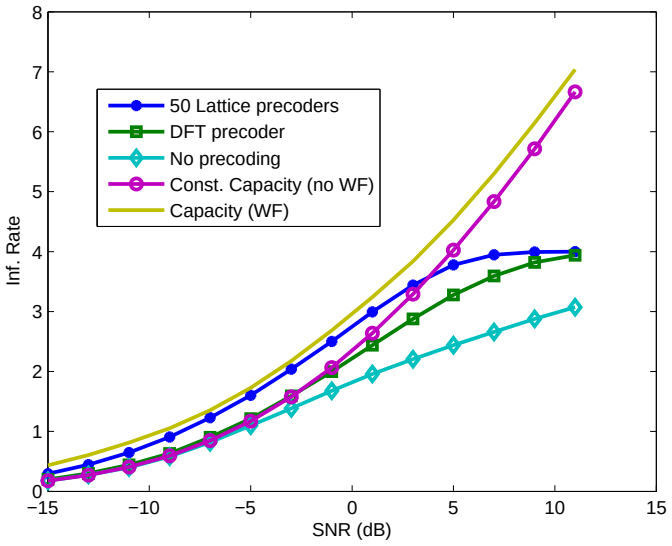

Figure 5.10: Information rate curves for $N = 4$ and binary signaling. Note how close the information rate of the lattice precoder codebook is to the capacity, even at low SNRs.

novel method to enumerate lattice precoders of degenerate rank, that turn off weak eigenmodes of the channel, is also presented. It is demonstrated by SER and information rate simulations that the obtained finite codebook performs very well.

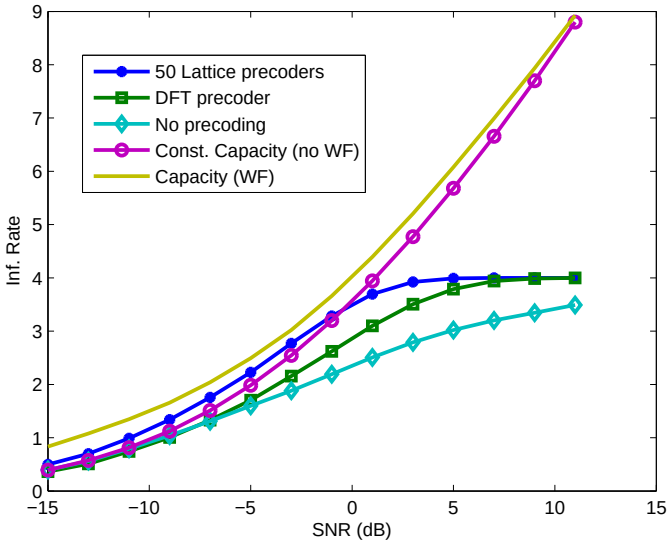

Figure 5.11: Information rate curves for $N = 4$ and binary signaling. Note how close the information rate of the lattice precoder codebook is to the capacity, even at low SNRs.

Chapter 6

Limited Feedback Precoding With MMSE Receiver

In this chapter, it is assumed that the transmitter (Tx) has no knowledge about the channel $\boldsymbol{H}$, while the receiver (Rx) has perfect knowledge of $\boldsymbol{H}$. The Tx CSI is more challenging to obtain and several levels of CSI can be assumed. The school of thought in this chapter is to assume a digital finite rate zero-error, zero-delay feedback link from the Rx to the Tx. A very popular method in this case is to have a finite set (codebook) of precoders that is known to both Rx and Tx, and to let Rx feed back the index of the precoder to use during a transmission. This is commonly referred to as limited feedback precoding. Unlike with perfect Tx CSI, the optimal codebook to use for this setup is not known to date.

Limited feedback for MIMO has been extensively studied in [103] - [111] and references therein. Performance metrics have been capacity (or outage probability) [105, 108], received SNR [104], minimum distance [112], BER [110], MMSE [107], etc. However, none of the aforementioned works study beamforming with full spatial multiplexing. Also they do not include power-loading, except for the work in [107, 108]. In order to facilitate analytic treatment of codebook designs, random beamforming codebooks have been considered in [106]. The outcome is that the random beamforming codebook approach performs well in the large system and codebook regime. Other works focus on methods to construct deterministic precoder codebooks with better performance [105, 107, 108, 109, 110, 111].

There are many ways to construct a codebook of precoders. One alternative is to constrain the codebook to unitary precoders (beamforming), i.e., precoders that are unitary matrices. One reason for this is that the peak to average power ratio (PAPR) at each transmit antenna is lower than for precoders with power-loading, hence, rendering it more amenable for practical implementation. The other precoding alternative is to have general precoding matrices subject to a power constraint (the unitary precoders immediately fulfill this power constraint). At first it may seem obvious that non-unitary precoding must be better than unitary since power-loading can boost the performance. By power-loading is meant that the columns in the precoder are not forced to have unit energy. Non-unitary precoding being better than unitary precoding is not necessarily true - if the precoding is unitary, then a re-enumeration operation of the antenna elements at the Rx can be done. The re-enumeration leads to gains in performance which could potentially be more significant than the power-loading gain. The main contribution of this chapter compared to prior work in the field is the observation and investigation of the antenna re-enumeration and to conduct a comparison between unitary and non-unitary precoding. A linear MMSE (Wiener filter) receiver is explicitly targeted. For this receiver structure, the focus is on the concept of Schur-convexity in [49] in order to construct good finite codebooks, which has not been explicitly addressed in the references. An additional contribution is a novel algorithm on constructing a unitary codebook based on the re-enumeration gain of the antenna elements. Moreover, it is shown by simulations that for rather small unitary codebooks, a random precoder codebook construction can perform as well as more advanced constructions. Additionaly, we present a method to construct a good non-unitary codebook that improves upon the random construction for smaller codebook sizes.

The chapter is organized as follows. Section 6.1 is split into three sections. Section 6.1.1 describes the optimal precoder for the case of perfect channel knowledge at Tx and Rx. Section 6.1.2 considers unitary precoding and introduces different codebook design methods for unitary precoders. Section 6.1.3 focuses on codebook design for non-unitary precoding. Section 6.2 presents receiver test results for the different schemes and discusses the results. Section 6.3 finally concludes the chapter.

6.1 Precoder Design

In Section 6.1.1, the structure of the optimal precoder for an MMSE receiver is described, given in [49]. Section 6.1.2 describes the unitary codebook design that is used later. Section 6.1.3 describes different methods of designing

codebooks where the precoders also perform power-loading.

6.1.1 Optimal precoder - perfect Tx CSI

The derivation of optimal precoders (denoted here as $\boldsymbol{F}_{\text{opt}}$) for perfect Tx CSI and MMSE receivers is given in [49] - a large number of different performance measures are treated. More precisely, the performance measures are divided into two parts: Schur-convex and Schur-concave functions of the MSE values. For Schur convex functions, the optimal precoder takes the form

$$\boldsymbol{F}_{\text{opt}} = \boldsymbol{V}\boldsymbol{D}_{\text{opt}}\boldsymbol{Q}, \tag{6.1}$$

where, as before, $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*$ is the SVD decomposition of $\boldsymbol{H}$, and the elements $d_{\text{opt},j}$ in the diagonal matrix $\boldsymbol{D}_{\text{opt}}$ are $d_{\text{opt},j} = \max(\mu/\sqrt{s_{j,j}} - 1/\sqrt{s_{j,j}}, 0)$ where μ is such that $\text{tr}(\boldsymbol{D}_{\text{opt}}^2) = N\,\text{SNR}$. $\boldsymbol{Q}$ is a unitary matrix where each element has the same magnitude. These matrices are often called Hadamard matrices in the literature.

Using $\boldsymbol{F}_{\text{opt}}$, the MSE matrix in (2.39) reduces to

$$\mathbb{E}\{(\hat{\boldsymbol{x}} - \boldsymbol{x})(\hat{\boldsymbol{x}} - \boldsymbol{x})^*\} = \boldsymbol{Q}^*(\boldsymbol{I} + \boldsymbol{D}_{\text{opt}}^2\boldsymbol{S}^2)\boldsymbol{Q}. \tag{6.2}$$

From the structure of $\boldsymbol{Q}$, it follows that the diagonal elements in E_{opt} (the MSE values) are equal. It is seen that the optimal precoder accomplishes this by multiplying out $\boldsymbol{V}$ from the channel and applying the $\boldsymbol{Q}$ matrix at the end. The power-loading matrix $\boldsymbol{D}_{\text{opt}}$ is used to minimize the sum of these equal MSE values, and is derived by solving a waterfilling problem in the singular values $\boldsymbol{S}$ of the channel. Hence the optimal precoder produces exactly the minimal solution of a Schur-convex function of the MSE's. Note that if no power-loading is used in the optimal precoder, i.e. $\boldsymbol{D}_{\text{opt}} = \boldsymbol{I}_{N\times B}$, then the resulting *unitary* precoder is simply making the MSE values equal, while not changing their total sum. Since a Schur-convex function of the MSE's is minimized when all the MSE values are equal, it is easily realized that this unitary precoder is the optimal precoder among all unitary precoders for Schur-convex functions of the MSE's. Its structure is

$$\boldsymbol{F}_{\text{opt,uni}} = \boldsymbol{V}\boldsymbol{Q}. \tag{6.3}$$

As shown in [49], the average BER is a Schur-convex function of the MSE values for a sufficiently high SNR (a rule of thumb given in [49] is an SNR such that BER $\leq 2 \cdot 10^{-2}$). Hence one optimal precoder for average BER is of the form given in (6.1), while one optimal unitary precoder is of the form in (6.3).

6.1.2 Limited feedback precoding - unitary codebook

The literature on unitary codebook precoder design for different performance measures is vast. The seminal work was done in [103] and [109], where it is shown that designing unitary codebooks is equivalent to packing unitary subspaces, which is known in the literature as Grassmanian codebook design. In [109], several different distance measures between the unitary subspaces were proposed with which the packing was performed. The different packings were produced with an algorithm with origins in [113]. In [114] a more systematic algorithm for constructing unitary precoders is presented, which utilizes the generalized Lloyd's algorithm and where the measures used in the algorithm are the ones from [109]. The ideas in [109] have also been applied to linear decoding. In [110], the authors use the chordal distance between subspaces together with Lloyd's algorithm to find good packings for the ZF receiver and the MMSE receiver. They also show that the unitary codebook should take into account the Hadamard matrix. More specifically, it is shown that in order to decrease BER, the optimal unitary precoder (assuming perfect CSI at Tx) should be of the form $\boldsymbol{V}_{\text{opt}} = \boldsymbol{V}\boldsymbol{Q}$ for some channel outcomes $\boldsymbol{H}$, where $\boldsymbol{Q}$ is a Hadamard matrix, while for some channel outcomes $\boldsymbol{Q}$ should not be included.

A problem with prior proposed codebooks is that they are only constructed for the case when not using full spatial multiplexing, i.e., when the unitary matrix is of dimension $N \times B$ and $B < N$. The distance metrics in [109] are meaningless for full spatial multiplexing as well as for subsequent work based on [109]. In [111], a way to construct a codebook of $N \times N$ unitary matrices is presented. This method will be implemented here for comparison. In [107], a distance measure between $N \times N$ unitary matrices is given, which is originally proposed in [113]. Hence the results derived herein should be compared to those in [107] as well. However, due to insufficient description of the codebook construction in [107], these results could not be replicated.

Next we look into the design methods for unitary codebook construction that will be compared in the numerical results section. First, a strengthened version of the method in [111] is developed. Assume a codebook $\mathcal{C}_{\text{uni}} = \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N\}$ of N unitary precoders. As discussed in the previous subsection, an optimal unitary precoder is of the form in (6.3), which gives the MSE matrix in (6.2) (where $\boldsymbol{D}_{\text{opt}} = \boldsymbol{I}_{N\times B}$). However, since we are limited to using $\mathcal{C}_{\text{uni}}$, the precoder is constructed as $\boldsymbol{F}_{\text{uni}} = \boldsymbol{U}_j\boldsymbol{Q}$. This gives the MSE matrix

$$\mathbb{E}\{(\hat{\boldsymbol{x}} - \boldsymbol{x})(\hat{\boldsymbol{x}} - \boldsymbol{x})^*\} = \boldsymbol{Q}^*\boldsymbol{U}_j^*\boldsymbol{V}(\boldsymbol{I} + \boldsymbol{S}^2)\boldsymbol{V}^*\boldsymbol{U}_j\boldsymbol{Q}. \tag{6.4}$$

Since an MSE matrix as in (6.2) is strived for, it is desirable to use a $\boldsymbol{U}_j$ such that $\boldsymbol{V}^*\boldsymbol{U}_j$ is close to the identity matrix, in which case (6.4) will be close to (6.2). This is the same as trying to approximate the precoder in (6.3) as close

as possible. However, since there is no power-loading matrix $\boldsymbol{D}$ as in (6.1), it is unnecessary to *only* strive for an optimal unitary precoder of the form in (6.3). As mentioned at the end of Section 6.1.1, the precoder in (6.3) is *one* optimal unitary precoder. $\boldsymbol{F}_{\text{opt,uni}}$ in (6.3) is such that the channel-precoder product is $\boldsymbol{H}\boldsymbol{F}_{\text{opt,uni}} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{I}\boldsymbol{Q}$, and the resulting MSE values will be equal. But they will also be equal if $\boldsymbol{H}\boldsymbol{F} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{I}^P\boldsymbol{Q}$, where $\boldsymbol{I}^P$ is a matrix obtained by permuting the columns of the identity matrix $\boldsymbol{I}$ in some way. Hence all the $N!$ possible permutations of $\boldsymbol{I}$ will yield MSE values that are equal. Thus, the corresponding precoders $\boldsymbol{F}$ give the same result and are all optimal. For large MIMO systems, $N!$ is large, and hence taking into account which permutation matrix that is approximated will considerably improve the performance. Note that if power-loading is present, this would often result in an unfavourable permutation of the power-loading, which would increase the MSE values and decrease the performance.

Denote all permutations by $\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_{N!}$. When there can be no ambiguity, we shall use $\boldsymbol{I}$ instead of $\boldsymbol{I}_1$. Figure 6.1 demonstrates the above discussed gain for the simple 1×1 MIMO case. The channel unitary matrix $\boldsymbol{V}$ is now simply a uniformly distributed vector on the unit circle. Assume that a codebook of two precoders $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ should be designed. The codebook should be designed such that the channel-precoder product $\boldsymbol{V}^*\boldsymbol{U}_j$ should be as close to the $\boldsymbol{I}_2$ axis as possible (note that the permutation matrix $\boldsymbol{I}_2$ corresponds to an axis for the simpe 1×1 MIMO case); this should hold for all $\boldsymbol{V}$. Clearly, one optimal placement of the two vectors is to place them as in Figure 6.1 in order to cover channel vectors that appear in one of the four regions marked by the dashed lines in the figure. All other optimal vectors are obtained by a rotation of $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$; in other words, the two optimal vectors must have a scalar product that is 0. However, the channel-precoder product can just as well be close to $\boldsymbol{I}_1$ (the other axis), which would simply lead to a rotation of the original channel-precoder product. Clearly $\boldsymbol{U}_1$ can be discarded because $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ cover the same space. Instead, it is then better to use the vector $\boldsymbol{U}$ together with $\boldsymbol{U}_2$, which gives a better coverage of the unit circle. In general, any two optimal vectors are separated by 45° (the scalar product is $1/\sqrt{2}$).

A similar argument can be applied to the 2×2 MIMO case. If $\boldsymbol{V}^*\boldsymbol{U}_j$ approximately equals $\boldsymbol{I}_2$ instead of $\boldsymbol{I}_1$, the physical interpretation is that the streams from the Tx have been permuted. The receiver can view this as a re-enumeration of the antenna elements, and the performance is the same as if $\boldsymbol{V}^*\boldsymbol{U}_j$ is close to $\boldsymbol{I}_1$. When only seeking for a channel-precoder product close to identity $\boldsymbol{I}$, the optimal two precoders are of the form $\boldsymbol{U}_1 = \boldsymbol{V}$, $\boldsymbol{U}_2 = \boldsymbol{V}\boldsymbol{I}_2$, where $\boldsymbol{V}$ is an arbitrary unitary matrix and $\boldsymbol{I}_2$ is obtained by swapping the two columns of the 2×2 identity matrix. The scalar product between $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$, defined as $|\text{tr}(\boldsymbol{U}_1^*\boldsymbol{U}_2)|$, is then 0. This corresponds to a packing as

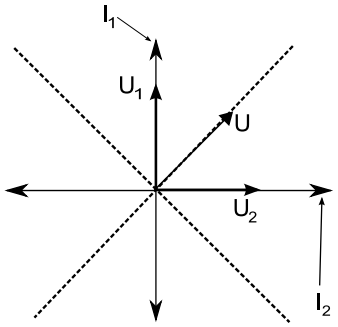

Figure 6.1: Packing vectors for the simple 1x1 MIMO case.

in Figure 6.1. By running the unitary precoder packing algorithm from [111], which will be explained shortly, we exactly get this solution, while Algorithm 1 below gives two matrices with scalar product equal to $\sqrt{2}$.

Next, a way to measure distance between a unitary matrix $\boldsymbol{Q}$ and a permutation matrix $\boldsymbol{I}_j$ is needed. Define the following function:

$$d_{\mathrm{u}}(\boldsymbol{X}, \boldsymbol{I}_j) \triangleq \| |\boldsymbol{X}| - \boldsymbol{I}_j \|. \tag{6.5}$$

Here $|\boldsymbol{X}|$ is the absolute value of each element in matrix $\boldsymbol{X}$. Observe that the function $d_{\mathrm{u}}(\cdot,\cdot)$ does not correspond to a valid metric. However, for notational convenience, (6.5) is refered to as the distance between $\boldsymbol{X}$ and $\boldsymbol{I}_j$. Plugging $\boldsymbol{X} = \boldsymbol{V}^*\boldsymbol{U}$ into (6.5) and some manipulations give

$$\begin{aligned} d_{\mathrm{u}}(\boldsymbol{V}^*\boldsymbol{U}, \boldsymbol{I}_j) &= \mathrm{tr}(|\boldsymbol{V}^*\boldsymbol{U}||\boldsymbol{V}^*\boldsymbol{U}|^* - |\boldsymbol{V}^*\boldsymbol{U}|(\boldsymbol{I}_j)^* \\ &\qquad -\boldsymbol{I}_j|\boldsymbol{V}^*\boldsymbol{U}|^* + \boldsymbol{I}) \\ &= 2N - \mathrm{tr}(|\boldsymbol{V}^*\boldsymbol{U}|(\boldsymbol{I}_j)^* + \boldsymbol{I}_j|\boldsymbol{V}^*\boldsymbol{U}|^*). \end{aligned} \tag{6.6}$$

From this form, it is seen that the distance measure in [111] is the same as in (6.6) for the special case when $\boldsymbol{I}_j = \boldsymbol{I}$. It is also seen that minimizing (6.5) is the same as maximizing the sum of the absolute values of the elements in $\boldsymbol{X}$ at positions indicated by the 1s in $\boldsymbol{I}_j$. Hence, (6.5) is an intuitive way to measure how close $\boldsymbol{X}$ is to $\boldsymbol{I}_j$, up to a rotation that does not affect the MMSE performance.

Assume we have a unitary codebook $\mathcal{C}_{\mathrm{uni}} = \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N\}$. For every uni-

tary matrix $\boldsymbol{U}_j$ in the codebook, define the set

$$\mathcal{T}_{\boldsymbol{U}_j,\boldsymbol{I}_k} = \{\boldsymbol{V} \quad : \quad d_{\mathrm{u}}(\boldsymbol{V}^*\boldsymbol{U}_j, \boldsymbol{I}_k) \leq d_{\mathrm{u}}(\boldsymbol{V}^*\boldsymbol{U}_l, \boldsymbol{I}_m), \\ l \neq j,\ m \neq k\}. \tag{6.7}$$

This is the set of all unitary matrices $\boldsymbol{V}$ that are closest to $\boldsymbol{I}_k$ if multiplied with $\boldsymbol{U}_j$ from the right. So every precoder $\boldsymbol{U}_j$ has $N!$ of these sets, each containing $\boldsymbol{V}$ that are closest to $\boldsymbol{U}_j$ and the corresponding permutation matrix. Based on these observations, it is possible to formulate a novel algorithm to construct a set $\mathcal{C}_{\mathrm{uni}} = \{\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N\}$ of N unitary matrices. Fix a positive integer p in advance. It has the setup of Lloyd's algorithm with clustering, and goes as follows:

Algorithm 1

1. (Initial): Put $j = 1$ and generate an initial set of N unitary matrices $\boldsymbol{U}_1^j, \boldsymbol{U}_2^j, \ldots, \boldsymbol{U}_N^j$.

2. (Clustering): Generate a new unitary matrix $\boldsymbol{V}$ from the channel. Find $(\boldsymbol{U}_m^j, \boldsymbol{I}_n) = \arg\min_{\boldsymbol{U}_k^j, \boldsymbol{I}_l} d_{\mathrm{u}}(\boldsymbol{V}^*\boldsymbol{U}_k^j, \boldsymbol{I}_l)$ and place $\boldsymbol{V}$ in the set $\mathcal{T}_{\boldsymbol{U}_m^j,\boldsymbol{I}_n}$. Repeat this step for many channel realizations.

3. (Discard): Keep only those sets $\mathcal{T}_{\boldsymbol{U}_m^j,\boldsymbol{I}_n}$ for which $n = p$ and discard the rest.

4. (Centroid): For each set $\mathcal{T}_{\boldsymbol{U}_m^j,\boldsymbol{I}_p}$, calculate the unitary centroid matrix $\boldsymbol{W}_{m,p}$ as
$$\boldsymbol{W}_{m,p} = \arg\min_{\hat{\boldsymbol{W}}_{m,p}} \mathbb{E}\{d_{\mathrm{u}}(\boldsymbol{V}^*\hat{\boldsymbol{W}}_{m,p}, \boldsymbol{I}_p)\} \tag{6.8}$$

5. (Update): Set $j{=}j{+}1$ and $\boldsymbol{U}_k^j = \boldsymbol{W}_{k,p}$. Return to 2.

The algorithm in [111] is similar to Algorithm 1, but there is no Discard step, and there is only one permutation matrix, namely the identity matrix $\boldsymbol{I}$. Hence, every precoder has only one set $\mathcal{T}_{\boldsymbol{U}_j}$ for each precoder $\boldsymbol{U}_j$. Some comments on Algorithm 1 are necessary. In the Clustering part, one simply generates a unitary matrix from the channel and places it in the set $\mathcal{T}_{\boldsymbol{U}_j,\boldsymbol{I}_k}$, i.e., to the unitary matrix $\boldsymbol{U}_j$ and permutation matrix $\boldsymbol{I}_k$ for which it comes closest to. In the Discard step, for every precoder $\boldsymbol{U}_j$, only one set is kept. In this case, it is decided to keep the same set for all precoders, hence the usage of the integer n. The Centroid step calculates a new unitary matrix that is closest (on average) to the unitary matrices in the set $\mathcal{T}_{\boldsymbol{U}_j,\boldsymbol{I}_k}$ by solving the

minimization problem in (6.8). Since it is very hard to find an exact solution, a suboptimal solution is constructed that follows the method in [111], and is outlined here. From (6.6), we see that solving (6.8) is equivalent to solving

$$\boldsymbol{W} = \arg \max_{\hat{\boldsymbol{U}}^* \hat{\boldsymbol{U}} = \boldsymbol{I}} \operatorname{tr}(|\boldsymbol{V}^* \hat{\boldsymbol{U}}| (\boldsymbol{I}_p)^* + \boldsymbol{I}_p |\boldsymbol{V}^* \hat{\boldsymbol{U}}|^*). \tag{6.9}$$

Put $\hat{\boldsymbol{U}} = \tilde{\boldsymbol{W}} \boldsymbol{I}_p$, where $\tilde{\boldsymbol{W}}$ is a unitary matrix. Then (6.9) becomes a maximization problem over $\tilde{\boldsymbol{W}}$

$$\begin{aligned} \hat{\boldsymbol{W}} &= \arg \max_{\tilde{\boldsymbol{W}}^* \tilde{\boldsymbol{W}} = \boldsymbol{I}} \operatorname{tr}(|\boldsymbol{V}^* \tilde{\boldsymbol{W}}| + |\boldsymbol{V}^* \tilde{\boldsymbol{W}}|^*) \\ &= \arg \max_{\tilde{\boldsymbol{W}}^* \tilde{\boldsymbol{W}} = \boldsymbol{I}} \sum_{k=1}^{N} |\boldsymbol{v}_k^* \tilde{\boldsymbol{w}}_k|, \end{aligned} \tag{6.10}$$

where $\boldsymbol{v}_k$ is the kth column in $\boldsymbol{V}$ and $\tilde{\boldsymbol{w}}_k$ the kth column in $\tilde{\boldsymbol{W}}$. Hence, the original solution $\boldsymbol{W}$ in (6.9) is obtained from the solution in (6.10) as $\boldsymbol{W} = \hat{\boldsymbol{W}} \boldsymbol{I}_p$. Thus, it is of interest to find a suboptimal solution to (6.10). By relaxing the constraint that $\tilde{\boldsymbol{W}}$ must have orthogonal columns to just having columns with unit length, it is seen from (6.10) that the minimization problem is then independent for each column $\tilde{\boldsymbol{w}}_k$, and thus each term in the summation in (6.10) can be maximized separately from the others. Hence, the following problem should be solved

$$\min_{\|\tilde{\boldsymbol{w}}_k\| = 1} \mathbb{E}\{|\boldsymbol{v}_{\boldsymbol{H},k}^* \tilde{\boldsymbol{w}}_k|^2\}. \tag{6.11}$$

It is simple to show that the solution to (6.11) is the eigenvector of the matrix $\mathbb{E}\{\boldsymbol{v}_{\boldsymbol{H},k} \boldsymbol{v}_{\boldsymbol{H},k}^*\}$ corresponding to the largest eigenvalue. Thus, there are N unit vectors $\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_N$, and we construct a matrix $\tilde{\boldsymbol{W}}_u$ with them. Since $\tilde{\boldsymbol{W}}_u$ is not a unitary matrix and thus not a solution to (6.10), simply let $\hat{\boldsymbol{W}}$ be the unitary matrix closest to $\tilde{\boldsymbol{W}}_u$ in Frobenius distance: $\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{Z}^* \boldsymbol{Z} = \boldsymbol{I}} \|\tilde{\boldsymbol{W}}_u - \boldsymbol{Z}\|$. The solution to this minimization problem is $\hat{\boldsymbol{W}} = \boldsymbol{X}\boldsymbol{Y}$, where $\tilde{\boldsymbol{W}}_u = \boldsymbol{X}\boldsymbol{\Sigma}\boldsymbol{Y}$, the SVD decomposition of $\tilde{\boldsymbol{W}}_u$.

The next strategy that is studied is random unitary precoding.

Algorithm 2

1. For every channel realization, generate randomly N unitary precoders $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N$.

For Algorithm 1 and 2, the receiver has the same selection criteria to find the best precoder in the codebook. The index I to feed back to Tx is determined as

Unitary Rx Selection Criteria

For each channel realization $\boldsymbol{H}$, find the feedback index B as

$$I = \arg\min_k \min_l d_{\mathrm{u}}(\boldsymbol{V}^*\boldsymbol{U}_k, \boldsymbol{I}_l). \tag{6.12}$$

For the algorithm in [111], the Rx selection criteria is the same except that $\boldsymbol{I}_l = \boldsymbol{I}$, $\forall l$, since only the distance to $\boldsymbol{I}$ is measured.

We close this section with a remark on the unitary precoders obtained from Algorithm 1 and 2. It will be demonstrated in Section 6.2 that a random codebook performs as well as the codebook obtained from Algorithm 1 for not so large codebook sizes. The reason for this is that there will be random samples of the unitary space that come as close to some permutation $\boldsymbol{I}_j$ (note that it does not matter which $\boldsymbol{I}_j$ it comes close to since the improved Unitary Rx Selection Criteria is used) as the codebook obtained from Algorithm 1. The simulation results in Section 6.2 show that a codebook of 16 precoders is sufficiently large to get the same performance of the two algorithms, which thus facilitates the codebook design of unitary matrices, since a random codebook construction is also efficient. However, for smaller codebooks, a packing gain is indeed obtained by the codebook resulting from Algorithm 1.

6.1.3 Limited feedback - non unitary precoding

In this section, we use Lloyd's algorithm to construct precoders that also perform power-loading. Due to the power-loading, the distance measure (6.5) needs to be replaced. As noted in Section 6.1, BER is a Schur-convex function of the MSE's for high SNR values. Hence it is desirable to minimize the MSE's. There are many different ways to measure the MSE values, but here the focus is on the sum of them, that is, the trace of the MSE matrix. Hence, the following metric is defined

$$d_{\mathrm{nu}}(\boldsymbol{H}, \boldsymbol{F}) = \mathrm{tr}((\boldsymbol{I} + \boldsymbol{F}^*\boldsymbol{H}^*\boldsymbol{H}\boldsymbol{F})^{-1}). \tag{6.13}$$

Other measures that are common are the product of the MSE values, the determinant of the MSE matrix, and also maximizing the smallest eigenvalue of the channel $\boldsymbol{H}$. These measures have been tested in Algorithm 3 below, but trace of the MSE matrix turned out to be superior. The scheme goes as follows:

Algorithm 3

1. (Initial): Put $j = 1$ and generate an initial set of N precoder matrices $\boldsymbol{F}_1^j, \boldsymbol{F}_2^j, \ldots, \boldsymbol{F}_N^j$.

2. (Clustering): Generate a new channel matrix $\boldsymbol{H}$.
   Find
   $$j = \arg\min_k d_{nu}(\boldsymbol{H}, \boldsymbol{F}^k)$$
   and place $\boldsymbol{H}$ in a set $\mathcal{R}_j$.

3. (Centroid): For each set $\mathcal{R}_j$, calculate the centroid channel matrix $\boldsymbol{H}_c^j$ as
   $$\boldsymbol{H}_c^j = \arg\min_{\boldsymbol{H}^j} \mathbb{E}_{\boldsymbol{H}\in\mathcal{R}_j}\{\|\boldsymbol{H} - \boldsymbol{H}^j\|\} \tag{6.14}$$
   Construct the new centroid precoder as $\boldsymbol{F}_c^j = \boldsymbol{V}_c^j \boldsymbol{D}_c \boldsymbol{Q}$, where $\boldsymbol{H}_c^j = \boldsymbol{U}_c^j \boldsymbol{S}_c (\boldsymbol{V}_c^j)^*$ and $\boldsymbol{D}_c$ is the waterfilling of the singular values $\boldsymbol{S}_c^j$ given in (6.1).

4. (Update): Put $j = j + 1$ and let $\boldsymbol{F}^j = \boldsymbol{F}_c^{j-1}$. Return to step 2.

This is a joint optimization over the power-loading and the unitary matrix in the precoders. The algorithm is simple and converges to a local optima after 3 – 4 iterations. In the Centroid step, the solution to (6.14) follows from Lemma 2.

Lemma 2. *The solution to* $\min_{\boldsymbol{H}_c} \mathbb{E}_H\{\|\boldsymbol{H} - \boldsymbol{H}_c\|\}$ *is* $\boldsymbol{H}_c = \mathbb{E}\{\boldsymbol{H}\}$.

Proof. Let $\mathbb{E}\{\boldsymbol{H}\} = \boldsymbol{H}_m$. It follows that

$$\begin{aligned}
&\mathbb{E}\{\|\boldsymbol{H} - \boldsymbol{H}_c\| = \mathbb{E}\{\mathrm{tr}((\boldsymbol{H} - \boldsymbol{H}_c)(\boldsymbol{H}^* - \boldsymbol{H}_c^*))\} \\
&= \mathbb{E}\{\mathrm{tr}((\boldsymbol{H} - \boldsymbol{H}_m + \boldsymbol{H}_m - \boldsymbol{H}_c)(\boldsymbol{H}^* - \boldsymbol{H}_m^* + \boldsymbol{H}_m^* - \boldsymbol{H}_c^*))\} \\
&\quad = \mathrm{tr}(\mathbb{E}\{(\boldsymbol{H} - \boldsymbol{H}_m)(\boldsymbol{H}^* - \boldsymbol{H}_m^*)\}) + \\
&\quad\quad \mathrm{tr}(\mathbb{E}\{(\boldsymbol{H}_m - \boldsymbol{H}_c)(\boldsymbol{H}_m^* - \boldsymbol{H}_c^*)\}) \\
&\quad\quad \geq \mathrm{tr}(\mathbb{E}\{(\boldsymbol{H} - \boldsymbol{H}_m)(\boldsymbol{H}^* - \boldsymbol{H}_m^*)\}),
\end{aligned} \tag{6.15}$$

with equality when $\boldsymbol{H}_m = \boldsymbol{H}_c$. □

Algorithm 3 was run for many different starting points, and interestingly enough, it converges to a set of precoders that have equal power-loading matrices. This effect is more and more evident when increasing the dimension of the system. Also, the sets $\mathcal{R}_j$ contain equally many channel matrices; hence, the precoders are uniformly placed in the space of $N \times N$ complex matrices, with respect to the distance measure in (6.13).

The next strategy studied is a random precoding codebook.

Algorithm 4: For every channel realization, randomly generate N precoders $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_N$. This is done by generating N random channel realizations

$\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N$. Then precoder $\boldsymbol{F}_j$ is constructed as $\boldsymbol{F}_j = \boldsymbol{V}_{\boldsymbol{H}_j} \boldsymbol{D}_j \boldsymbol{Q}$, where $\boldsymbol{D}_j$ is the waterfilling solution to the singular values in channel $\boldsymbol{H}_j$.

The receiver has the following selection criteria for Algorithm 3 and 4.

Non-unitary Rx Selection Criteria: For each channel realization $\boldsymbol{H}$, find the feedback index B as $B = \arg\min_k d_{nu}(\boldsymbol{H}, \boldsymbol{F}_k)$.

6.2 Numerical Results

In this section we present SER simulations over Rayleigh fading MIMO channel gains for the different algorithms. They are compared to each other, and also with the algorithm proposed in [111], explained in the previous section. No simulation results for Algorithm 2 are included, since it turns out that using precoders produced by Algorithm 2 together with the Rx Selection Criteria (6.12), the same SER performance is achieved as when using precoders resulting from Algorithm 1 or the precoders from [111]. This behavior is expected for a sufficiently large codebook as was discussed in Section 6.1.2, and the simulation figures in this section show that a codebook of only 16 precoders is enough to get the same performance. Hence, in this case, performance gains arise only from the unitary Rx selection criteria in (6.12), since it is a method that improves *any* unitary codebook. This is an interesting result, since it shows that a random unitary codebook construction is a good method to construct unitary codebooks. For codebook sizes smaller than 16, Algorithm 1 achieves an improved packing over a random unitary codebook.

All the figures present SER simulations for different MIMO systems, codebook sizes and data alphabets. In the figure legends, "Algorithm in [111]" refers to precoders constructed by the algorithm in [111] and that use the Rx selection criteria therein, explained right after (6.12). The following can be concluded from the figures. The results differ for different alphabet sizes, so first we consider 4-QAM. For 4-QAM, Figures 6.2 - 6.8 show that there is a clear gain (about 0.5 dB) to use Unitary Rx Selection Criteria (6.12) than using the selection criteria in [111]. Also, the performance is significantly better than using no precoding. When it comes to non-unitary precoding, for a small codebook size, precoders from Algorithm 3 significantly outperform precoders generated by Algorithm 4 (around 0.5 dB). However, when the codebook size is increased (to 128 or 256 precoders), Algorithm 4 performs as well as Algorithm 3. As noted in Section 6.1.2, it has also been mentioned in [110] that random precoding performs well. However, that was for unitary precoding, while the figures here show that this also holds for non-unitary precoding.

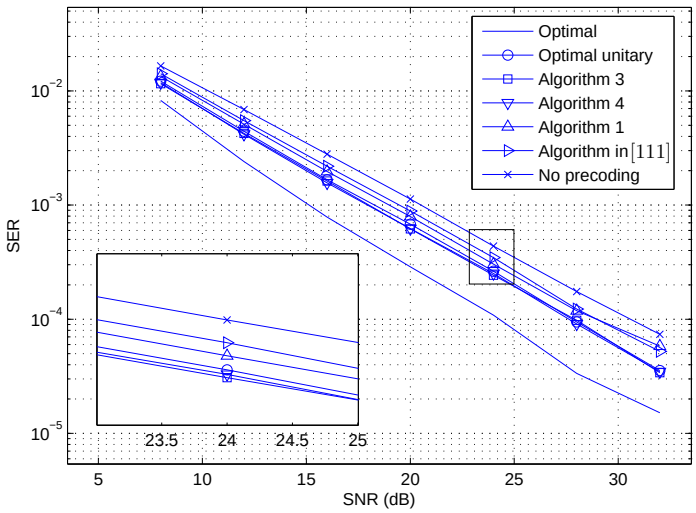

Figure 6.2: 3×3 MIMO, 256 precoders and 4-QAM.

What is also important to note is that unitary and non-unitary precoding have close to equal performance for a small codebook size (16 precoders or less). When increasing the codebook size, non-unitary precoding performs better than the optimal unitary precoder; hence codebooks with power-loading are very beneficial when it comes to SER.

For 16-QAM, the story is a bit different. Figure 6.8 shows that the gain from using unitary precoding compared to no precoding is not significant anymore (not even by using optimal unitary precoding). There is a clear improvement in using power-loading codebooks. Also, Algorithm 4 performs at least as well as Algorithm 3 for a large codebook size, while for a smaller codebook (16 precoders), Algorithm 3 is better than 4.

6.3 Conclusion

The performance of limited feedback precoding with unitary and non-unitary precoder codebooks and an MMSE receiver was investigated. Different algorithms to construct the two types of codebooks were investigated and compared with each other. Unitary precoding has an inherent gain by having the freedom to re-enumerate the streams at Rx and by this improve the MSE values. This is possible for *any* unitary codebook and is a significant gain compared to

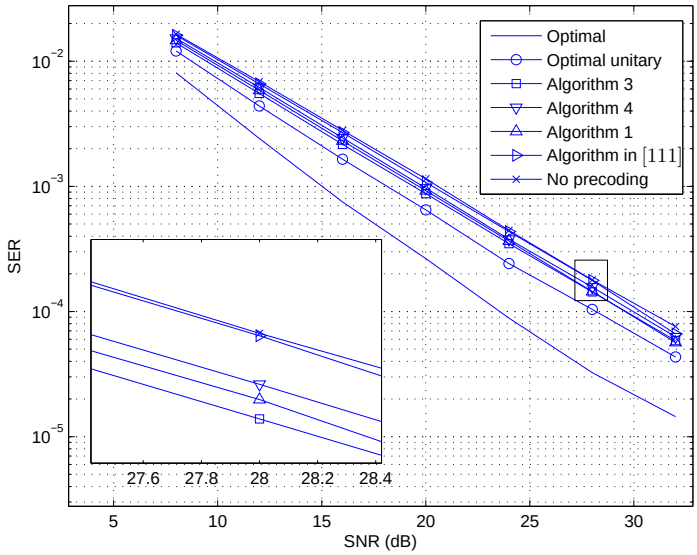

Figure 6.3: 3×3 MIMO, 16 precoders and 4-QAM.

the selection method in [111] as demonstrated in the simulations. Moreover, it has been argued and shown by simulations that random unitary precoders perform as well as more advanced constructions. The simulations also show that non-unitary codebooks significantly outperform unitary ones for large codebook sizes, while for smaller sizes, the performance is the same. Also, in the large codebook regime, random non-unitary precoding performs as well as non-unitary precoding steming from a simple packing algorithm, while for smaller codebook sizes, the packing has a significant gain. The non-unitary packing exhibits an interesting structure: the resulting precoders have the same power-loading. For larger alphabet sizes, unitary precoding yields small performance gains, while non-unitary precoding still performs very well.

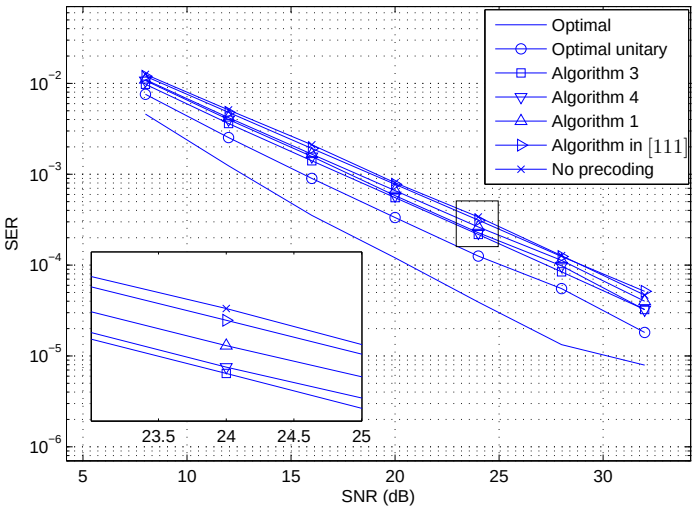

Figure 6.4: 4×4 MIMO, 128 precoders and 4-QAM.

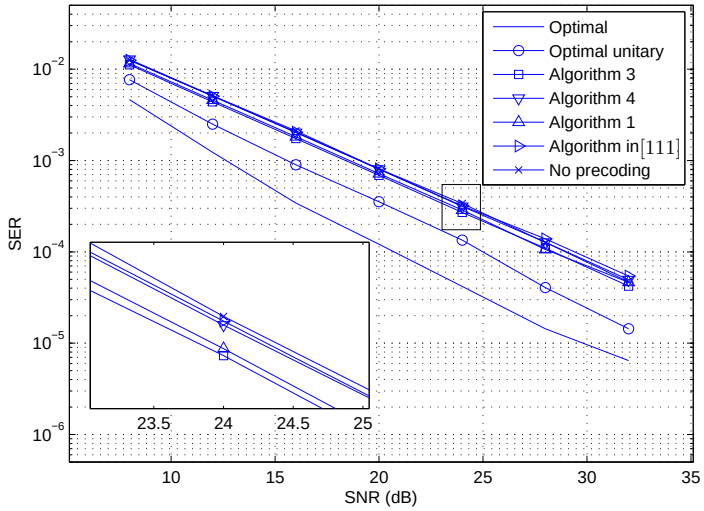

Figure 6.5: 4×4 MIMO, 16 precoders and 4-QAM.

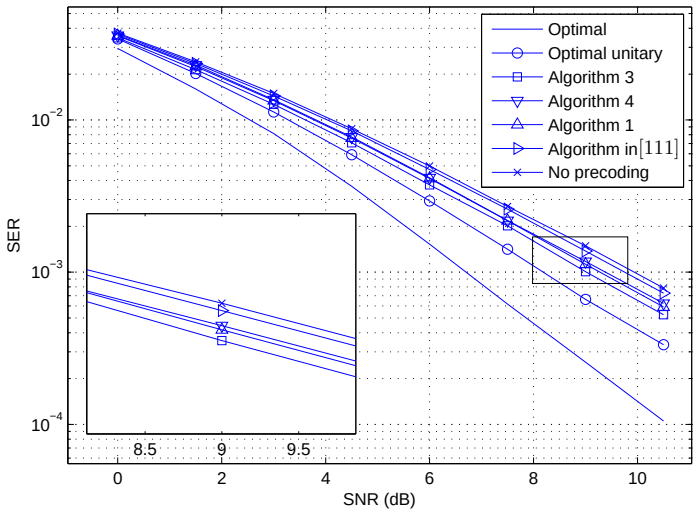

Figure 6.6: 3×4 MIMO, 16 precoders and 4-QAM.

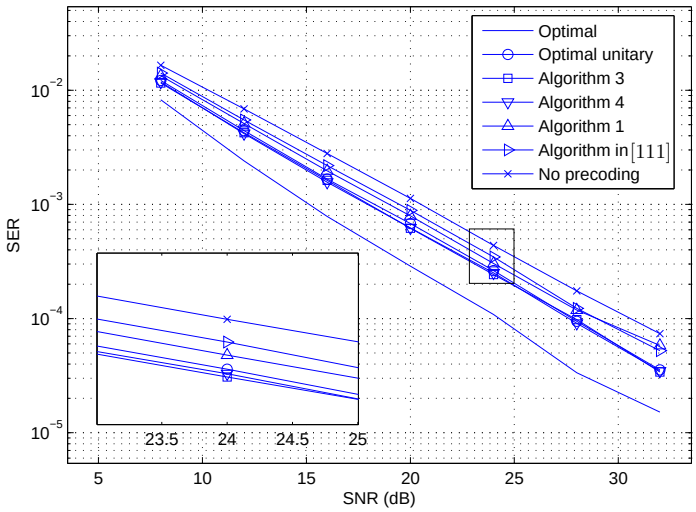

Figure 6.7: 3×4 MIMO, 256 precoders and 4-QAM.

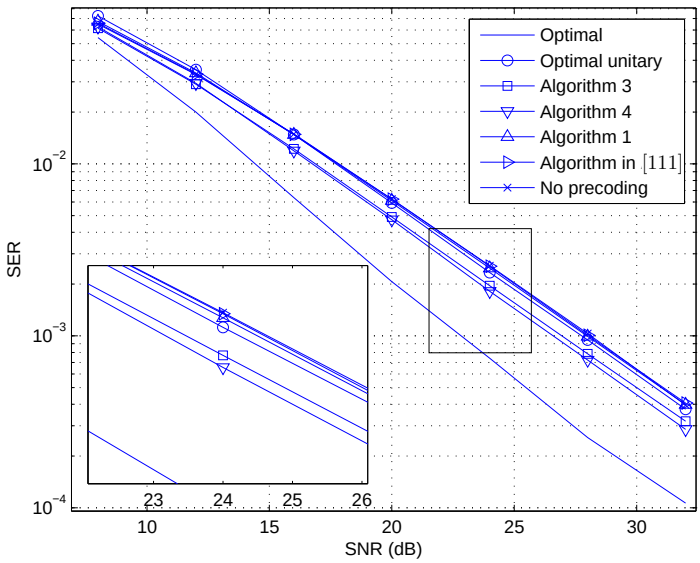

Figure 6.8: 4×4 MIMO, 128 precoders and 16-QAM.

Chapter 7

Future Work

The main result in Part II is the insight on the behavior of linear precoders that maximize the minimum distance between the received signaling points over linear channels. This is something that has been a subject of research for many decades, and is also of interest for practical applications. Part II in this thesis completely explains the behavior of these linear precoders once the signaling alphabet becomes large. Among other things, it shows that construction of optimal minimum distance precoders is in essence a *discrete* optimization problem for large alphabets. It is however well-known from before that this problem is NP-hard to solve. Moreover, even if it is discrete, it becomes untractable for larger dimensions: Enumerating all possible solution candidates is essentially impossible, even if it is done off-line. Hence, future work should consider efficient suboptimal solutions to this problem, that still have acceptable performance. The work in Part II shows that the main challenge in constructing the optimal precoder for dimensions N up to 8, for a certain channel, is equivalent to finding the optimal basis vectors (i.e., the optimal unimodular $\boldsymbol{Z}$ matrix) of a perfect lattice: If this basis would somehow be known before hand, constructing the optimum precoder for small enough dimension is then a rather easy task. Thus, a possible future direction is to get good estimates on $\boldsymbol{Z}$. It has already been shown in Section 4.3.7 that the GMD precoder produces a good estimate of $\boldsymbol{Z}$, but obtaining this $\boldsymbol{Z}$ from the GMD precoder is also an NP-hard problem. Hence, other suboptimal constructions of $\boldsymbol{Z}$ are therefore needed. One possible method is the iterative optimization presented in [79], that now could be used in combination with perfect lattices.

Beside the large alphabet limits, the work in Part II shows that lattice precoders also perform very well for small alphabets. The problem is in a way divided into finding precoders of full rank and degenerate rank. For full rank

precoders, the problem is again about finding vertices in a polytope, but now in a finite Ryshkov polytope. It was observed by running the `lrs` software in [97], that these vertices are still perfect lattices even for finite alphabets. However, this does not have to hold in higher dimensions, and future work should focus on classifying the vertices of the finite Ryshkov polytope, to see whether some of its vertices represent non-perfect lattices. For degenerate rank precoders, a novel method was developed in Part II that finds lattice precoders of degenerate rank. It has been demonstrated through simulation that it is important to include these precoders into the finite codebook, since they avoid transmission across weak eigenmodes. Even though these precoders can be found off-line, just as the full rank vertices, this becomes a more complex task for larger alphabets (note however that the larger the alphabet, the less these precoders will occur as optima). Therefore, it is again of need to find good suboptimal constructions even for this case.

As demonstrated by the information rate simulations, the minimum distance precoders converge quickly to the maximum information rates with increasing SNR. Moreover, it was seen that for some channels, they perform very close to the capacity even in the low SNR regime. More specifically, there are certain channels for which the lattice precoders reach the capacity at low SNRs. Characterization of these channels is of importance, since that could give insight in how to close the gap to the capacity even for channels where this currently does not happen. A possible method of approach is to combine the strength of the minimum distance precoders in the high SNR regime, together with the strength of the Mercury/Waterfilling technique at low SNRs. This could result in efficient precoding methods that come very close to capacity even for small alphabets.

References

[1] A. G. Lillie, A. P. Miguelez, A. R. Nix and J. P. McGeehan, "A comparison of multi-carrier OFDM and single carrier iterative equalisation for future high performance wireless local area networks", in *Proc. IEEE Vehicular Tech. Conf. (VTC)*, Vancouver, BC, Sep. 2002.

[2] L. Zhu, J. Zhang, Y. Pei, N. Ge and J. Lu, "On maximum achievable information rates of single-carrier and multi-carrier systems over the ultra wideband channels", in *Proc. IEEE International Conference on Ultra-Wideband (ICUWB)*, Nanjing, China, Sep, 2010.

[3] N. Benvenuto, R. Dinis, D. Falconer and S. Tomasin, "Single carrier modulation with nonlinear frequency domain equalization: An idea whose time has come-again," *Proceedings of the IEEE*, Vol. 98, No. 1, pp. 69–96, Jan. 2010.

[4] T. Shi, S. Zhou and Y. Yao, "Capacity of single carrier systems with frequency-domain equalization," in *Proc. IEEE 6th CAS Symp.*, Shanghai, China, pp. 429–432, May 2004.

[5] L. Ye and A. Burr, "Frequency diversity comparison of coded SC-FDE and OFDM on different channels," in *Proc. Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, Athens, Greece, Sep. 2007.

[6] J. Tubbax, et. al, " OFDM versus single carrier: A realistic multi- antenna comparison", *EURASIP Journal on Applied Signal Processing* no. 9, pp. 1275–1287, 2004.

[7] I. Kaya, K. Turk and Y. Baltacy, "Experimental BER performance evaluation of OFDM and single carrier transmissions in real-time Wimax radio", in *Proc. Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, Athens, Greece, Sep. 2007.

[8] E. Dahlman, S. Parkvall, J. Sköld, and P. Beming, *3G evolution: HSPA and LTE for mobile broadband*, 2nd ed., Elsevier Science, Sep. 2008.

[9] J. G. Proakis and M. Salehi, *Digital Communications,* 5th ed., McGraw-Hill, NY, 2008.

[10] G. Ungerboeck, "Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems," *IEEE Trans. Commun.*, vol. 22, no. 5, pp. 624–636, May 1974.

[11] G. D. Forney Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. 18, no. 2, pp. 363–378, May 1972.

[12] H. Nyquist, "Certain factors affecting telegraph speed", *Bell System Technical Journal*, 3:324 – 346, Apr. 1924.

[13] C. E. Shannon, "A mathematical theory of communications," *Bell System Technical Journal*, 27:379–429 and 27:623–656, Jul. and Oct. 1948.

[14] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, 23:10–21, 1949.

[15] Y. G. Yoo and J. H. Cho, "Asymptotic optimality of binary faster-than-Nyquist signaling", *IEEE Commun. Letters*, vol. 14, no. 9, pp. 788–790, Sep. 2010.

[16] S. Shamai, L. H. Ozarow and A. D. Wyner, "Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. Inf. Theory*, vol. 37, no. 6, pp. 1527–1539, Nov. 1991.

[17] W. Hirt, *Capacity and information rates of discrete-time channels with memory,* Ph.D thesis, no. ETH 8671, Inst. Signal and Information Processing, Swiss Federal Inst. Technol., Zurich, 1988.

[18] F. Rusek and J. B. Anderson, "Constrained capacities for faster-than-Nyquist signaling, *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 764–775, Feb. 2009.

[19] R. A. Gibby and J. W. Smith, "Some extensions of Nyquist's telegraph transmission theory", *Bell System Technical Journal*, 44(2):1487–1510, Sep. 1965.

[20] M. Abramowitz and I. A. Stegun, (Eds.). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th printing, New York: Dover, p. 11, 1972.

[21] J. B. Anderson, *Digital Transmission Engineering*, IEEE Press, Piscataway, NJ, 2nd ed., 2005.

[22] T. Starr, J. M. Cioffi and P. J. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice Hall, Upper Saddle River, NJ 1999.

[23] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[24] A. Paulraj, R. Nabar and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, Cambridge, UK, 2003.

[25] C. B. Peel, B. M. Hochwald and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna communication part I: Channel inversion and regularization," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 195–202, Jan. 2005.

[26] L.-U. Choi and R. D. Murch, "A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 20–24, Jan. 2004.

[27] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna communication part II: Perturbation," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 537–544, May 2005.

[28] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian MIMO broadcast channel," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Chicago, Jun.-Jul. 2004.

[29] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.

[30] L. Zheng and D. N. C. Tse, "Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 359–383, Feb. 2002.

[31] R. M. Gray, *Toeplitz and Circulant Matrices: A review*, Foundations and Trends in Communications and Information Theory, NOW Publishers.

[32] R. W. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission," *Bell System Technical Journal*, 46:1775–1796, 1966.

[33] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE Trans. Commun. Tech.*, vol. 19, no. 5, pp. 628–634, Oct. 1971.

[34] F. Rusek and A. Prlja, "Optimal channel shortening for MIMO and ISI channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 810–818, Feb. 2012.

[35] J. Salz, "Digital transmission over cross-coupled linear channels," *AT&T Technical Journal*, vol. 64, no. 6, pp. 1147–1159, Jul.-Aug. 1985.

[36] G. G. Raleigh and J. M. Cioffi, "Spatio-temporal coding for wireless communication," *IEEE Trans. Commun.*, vol. 46, no. 3, pp. 357–366, Mar. 1998.

[37] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multiple antennas," *Bell System Technical Journal*, pp. 41–59, Autumn 1996.

[38] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, Kluwer Academic Publishers, vol. 6, no. 3, pp. 311–335, 1998.

[39] I. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Tel.*, vol. 10, no. 6, pp. 585–595, Nov.-Dec. 1999.

[40] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. Int. Symp. Turbo Codes*, pp. 1–11, ENST de Bretagne, France, Sep. 1997.

[41] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, Inc., Hoboken, NJ, 2006.

[42] S. Brink, G. Kramer and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[43] B. Lu, G. Yue and X. Wang, "Performance analysis and design optimization of LDPC-coded MIMO OFDM systems," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 348-361, Feb. 2004.

[44] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 549-583, Feb. 2006.

[45] E. G. Larsson, "MIMO detection methods: How they work," *IEEE Sig. Process. Mag.*, vol. 26, no. 3 , pp. 91–95, May 2009.

[46] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple antenna channel," *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 389–399, Mar. 2003.

[47] N. Merhav, G. Kaplan, A. Lapidoth and S. Shamai, "On information rates for mismatched decoders," *IEEE Trans. Inf. Theory*, Nov. 1994.

[48] A. Ganti, A. Lapidoth and I. E. Telatar, "Mismatched decoding revisited: General alphabets, channels with memory, and the wide-band limit," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2315-2328, Nov. 2000.

[49] D. P. Palomar and Y. Jiang, *MIMO transceiver design via majorization theory*, Foundations and Trends in Communication and Information Theory, NOW Publishers, vol. 3, nos. 4–5, 2007.

[50] D. P. Palomar, J. M. Cioffi and M. A. Lagunas, "Joint Tx-Rx beamforming design for multicarrier MIMO channels: A unified framework for convex optimization," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2381–2401, Sep. 2003.

[51] D. J. Ryan, I. B. Collings, I. V. L. Clarkson and R. W. Heath, "Performance of vector perturbation multiuser (MIMO) systems with limited feedback," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2633–2644, Sep. 2009.

[52] A. Razi, D. J. Ryan, I. B. Collings and J. Yuan, "Sum rates, rate allocation, and user scheduling for multi-user MIMO vector perturbation precoding", *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 356–365, Jan. 2010.

[53] K. Kusume, M. Joham, W. Utschick and G. Bauch, "Efficient Tomlinson-Harashima precoding for spatial multiplexing on flat MIMO channel," in *Proc. IEEE Int. Conf. Comm. (ICC)*, Seoul, South Korea, pp. 2021–2025, May 2005.

[54] E. Dahlman, S. Parkvall and Johan Sköld, *4G LTE/LTE-Advanced for Mobile Broadband*, Academic Press (London), 2011.

[55] J. He and M. Salehi, "A lattice precoding scheme for flat-fading MIMO channels,"in *Proc. IEEE MILCOM*, San Diego, CA, Nov. 2008.

[56] C. Xiao, Y. R. Zheng and Z. Ding, "Globally optimal linear precoders for finite alphabet signals over complex vector Gaussian channels," *IEEE Trans. Signal Process.,* vol. 59, no. 7, pp. 3301–3314, Jul. 2011.

[57] S. S. Lokesh, A. Kumar and M. Agrawal, "Structure of an optimum linear precoder and its application to ML equalizer," *IEEE Trans. Signal Process.*, vol. 56, no. 8, Aug. 2008.

[58] M. Payaro and D. P. Palomar, "On optimal precoding in linear vector Gaussian channels with arbitrary input distribution," in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Seoul, Jun. 2009.

[59] H. Lee, S. Park and I. Lee, "A new MIMO beamforming technique based on rotation transformations," in *Proc. IEEE Int. Conf. Comm. (ICC)*, Glasgow, Scotland, Jun. 2007.

[60] M. Vu and A. Paulraj, "Optimal linear precoders for MIMO wireless correlated channels with nonzero mean in space-time coded systems,", *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2318–2332, Jun. 2006.

[61] A. Scaglione, P. Stoica, S. Barbarossa, G. B. Giannakis and H. Sampath, "Optimal designs for space-time linear precoders and decoders," *IEEE Trans. Signal Process.,* vol. 50, no. 5, pp. 1051–1064, May 2002.

[62] S. K. Mohammed, E. Viterbo, Y. Hong and A. Chockalingam, "MIMO precoding with X- and Y-codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3542–3566, Jun. 2011.

[63] N. Jindal, "MIMO broadcast channels with finite-rate feedback," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5045–5060, Nov. 2006.

[64] J.B. Anderson and A. Svensson, *Coded Modulation Systems*, Plenum, NY, 2003.

[65] B. Vrigneau *et al.*"Extension of the MIMO precoder based on the minimum Euclidean distance: A cross-form matrix," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 2, pp. 135–146, Apr. 2008.

[66] Q.-T. Ngo, O. Berder and P. Scalart, "General minimum Euclidean distance based precoder for MIMO wireless systems," accepted for publication in *EURASIP Journal on Advances in Signal Processing.*

[67] L. Collin, O. Berder, P. Rostaing and G. Burel, "Optimal minimum-distance based precoder for MIMO spatial multiplexing systems," *IEEE Trans. Signal Process.*, vol. 52, no. 3, pp. 617–627, Mar. 2007.

[68] R. Fletcher. *Practical Methods of Optimization*, 2nd ed. Wiley- Interscience, New York, 1987.

[69] T. Abrudan, J. Eriksson and V. Koivunen, "Conjugate gradient algorithm for optimization under unitary matrix constraint", *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 1704–1714, Sep. 2009.

[70] T. Abrudan, J. Eriksson and V. Koivunen, "Steepest descent algorithms for optimization under unitary matrix constraint," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1134–1147, Mar. 2008.

[71] A. Said and J.B. Anderson, "Design of optimal signals for bandwidth-efficient linear coded modulation", *IEEE Trans. Inf. Theory*, vol. 44, pp. 701–713, Mar. 1998.

[72] Q. T. Ngo, O. Berder, B. Vrigneau and O. Sentieys, "Minimum distance based precoder for MIMO-OFDM systems using a 16-QAM modulation", in *Proc. IEEE Int. Conf. Comm. (ICC)*, Dresden, Germany, Jun. 2009.

[73] Q. T. Ngo, O. Berder and P. Scalart, "3-D minimum Euclidean distance based sub-optimal precoder for MIMO spatial multiplexing systems," in *Proc. IEEE Int. Conf. Comm. (ICC)*, Cape Town, South Africa, May 2010.

[74] J. H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York 1999.

[75] F. A. Monteiro and I. J. Wassell, "Recovery of a lattice generator matrix from its Gram matrix for feedback and precoding in MIMO," in *Proc. 4th International Symposium on Communications, Control and Signal Processing*, Limassol, Cyprus, Mar. 2010.

[76] M. Szydlo, "Hypercubic lattice reduction and analysis of GGH and NTRU signatures," in *Proc. Eurocrypt 2003*, Warsaw, Poland, 2003.

[77] R. Helaman, P. Ferguson and D. H. Bailey, "A polynomial time, numerically stable integer relation algorithm," *RNR Technical Report RNR-91-032*, Jul. 1992.

[78] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, pp 463–471, Apr. 1985.

[79] S. Bergman, *Bit loading and precoding for MIMO communication systems*, Ph.D. thesis, Signal Processing Laboratory, Royal Institute of Technology, Stockholm, May 2009.

[80] C.F. Gauss, Disquisitiones Arithmeticae. Leipzig 1801. German translation: Untersuchungen über die hohere Arithmetik. Springer, Berlin 1889. (reprint: Chelsea, New York, 1981.)

[81] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, Taipei, Nov. 2002.

[82] A. Lozano, A. M. Tulino and S. Verdu, "Mercury/waterfilling: Optimum power allocation with arbitrary input constellations", *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3033–3051, Jul. 2006.

[83] F. Perez-Cruz, M. R. D. Rodrigues and S. Verdu, "MIMO Gaussian channels with arbitrary inputs: Optimal precoding and power allocation," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1070–1084, Mar. 2010.

[84] A. Schürmann, *Computational geometry of positive definite quadratic forms*, AMS, Providence, 2008.

[85] J. Martinet, *Perfect Lattices in Euclidean Spaces*, Springer-Verlag, Heidelberg, 2003.

[86] H. Minkowski, "Diskontinuitätsbereich für arithmetische äquivalenz", *J. Reine Angew. Math.*, vol. 129 (1905), pp. 220–274, Reprint in Gesammelte Abhandlungen, Band II, Teubner, Leipzig, 1911.

[87] S.S. Ryshkov, "The polyhedron $\mu(m)$ and certain extremal problems of the geometry of numbers", *Soviet Math. Dokl.* 11 (1970), pp. 1240–1244, translation from Dokl. Akad. Nauk SSSR 194, pp. 514–517 (1970).

[88] M. J. Cohn, Z. D. Lomakina and S. S. Ryshkov, "Vertices of the symmetrized Minkowski region for $n \leq 5$", *Proc. Steklov Inst. Math.*, vol. 152 (1982), pp. 213–223, translation from Tr. Mat. Inst. Steklova 152, pp. 195–203 (1980).

[89] G. D. Forney Jr. and L.-F. Wei, "Multidimensional constellations. I. Introduction, figures of merit, and generalized cross constellations," *IEEE Journal on Selected Areas Commun.*, vol. 7, no. 6, pp. 877–892, Aug. 1989.

[90] P. M. Gruber and J. M. Wills, *Handbook of Convex Geometry, vol. A*, Elsevier Science Publishers B.V., Amsterdam, Netherlands 1993.

[91] W. H. Mow, "Universal lattice decoding: Principle and recent advances," *Wireless Communications and Mobile Computing*, Special Issue on Coding and Its Applications in Wireless CDMA Systems, vol. 3, no. 5, pp. 553-569, Aug. 2003.

[92] Y. Jian, J. Li and W. W. Hager, "Uniform channel decomposition for MIMO communications", *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4283–4294, Nov. 2005.

[93] J. Milnor and D. Husemoller, *Symmetric Bilinear Forms*, Springer, 1973.

[94] D. L. Kleinman and M. Athans, "The design of suboptimal linear time-varying systems," *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 150–159, Apr. 1968.

[95] G. F. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier Mémoire. Sur quelques propriétés des formes quadratiques positives parfaites," *J. Reine Angew. Math.*, vol. 133, pp. 97–178, 1907.

[96] M. M. Gutzmann, O. Preusche, C. Rahn and W. Erhard, "Efficiently Enumerating Unimodular Mappings," *Berichte zur Rechnerarchitektur*, Technical Report, vol. 3, no. 10, 1997.

[97] D. Avis, "LRS: A revised implementation of the reverse search vertex enumeration algorithm," *Polytopes - Combinatorics and Computation*, G. Kalai & G. Ziegler eds., Birkhauser-Verlag, DMV Seminar Band 29, pp. 177–198, 2000.

[98] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, Kluwer Academic Publishers, 2002.

[99] www2.research.att.com/ njas/lattices.

[100] D. Kapetanović and F. Rusek, "Linear Precoders for Parallell Gaussian Channels with Low Decoding Complexity", in *Proc. IEEE Vehicular Tech. Conf.*, San Francisco, Calif., Sep. 2011.

[101] S. P. Boyd and L. Vandenberghe, *Convex Optimization (pdf)*, Cambridge University Press, 2004.

[102] G. H. Golub, L. Van and Charles F, *Matrix Computations* 3rd ed., Johns Hopkins Studies in Mathematical Sciences, 1996.

[103] D. Agrawal, T. J. Richardson and R. Urbanke, "Multiple-antenna signal constellations for fading channels", in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, Sorrento, Italy, Jun. 2000.

[104] X. Penefei and G. B. Giannakis, "Design and analysis of transmit-beamforming based on limited-rate feedback", in. *Proc. IEEE Vehicular Tech. Conf. (VTC)*, Los Angeles, CA, Sep. 2004.

[105] J. C. Roh and B. D. Rao, "Design and analysis of MIMO spatial multiplexing systems with quantized feedback," *IEEE Trans. Signal Process.*, vol. 54, no. 8, pp. 2874–2886, Aug. 2006.

[106] W. Santipach and M. L. Honig, "Capacity of a multiple-antenna fading channel with a quantized precoding matrix", *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1218–1234, Mar. 2009.

[107] Y. Liu and H. Zhang, " Beamforming for MIMO systems with limited feedback", *IET Intl. Conf. Wireless, Mobile and Multimedia Networks*, Nov. 2006.

[108] K. K. Mukkavilli, A. Sabharwal and B. Aazhang, "Generalized beamforming for MIMO systems with limited transmitter information," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2003.

[109] D. J. Love and R. W. Heath, "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2967–2976 , Aug. 2005.

[110] S. Zhou and B. Li, "BER criterion and codebook construction for finite-rate precoded spatial multiplexing with linear receivers", *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1653–1665, May 2006.

[111] E. Sengul, H. J. Park and E. Ayanoglu, "Bit-interleaved coded multiple beamforming with imperfect CSIT," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1505–1513, May 2009.

[112] A. Ghaderipoor and C. Tellambura, "Minimum distance-based limited-feedback precoder for MIMO spatial multiplexing systems", in *Proc. IEEE Vehicular Tech. Conf. (VTC)*, Montreal, Canada, Sep. 2006.

[113] J. H. Conway, R. H. Hardin and N. J. A. Sloane "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Exper. Math.*, vol. 5, pp. 139–159, 1996.

[114] A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, Norwell, MA: Kluwer Academic, 1992.

[115] J. Nie, K. Ranestad and B. Sturmfels, "The algebraic degree of semidefinite programming," *Mathematical Programming*, vol. 122, no. 2, pp. 379-405, 2010.