The Design of Linear Space-Time Codes For Quasi-Static Flat-Fading Channels

A Thesis
Presented to
The Academic Faculty
By

Badri Varadarajan

In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Electrical Engineering

School of Electrical and Computer Engineering

Georgia Institute of Technology

July 2004

The Design of Linear Space-Time Codes For Quasi-Static Flat-Fading Channels

Approved:

JOHN R. BARRY, Chairman

GORDON L. STUBER

YE (GEOFFREY) LI

Date Approved ___July 7, 2004___

Acknowledgments

In the second year of my Ph.D., I met my advisor Dr. John Barry almost every week. In every meeting, I would enthusiastically describe a new problem I had found, and my own oh-so-wonderful ideas. Through it all, Dr. Barry listened patiently, and always seemed to know both the history of each problem and the state of the art. Finally, he filtered one of my wild ideas and introduced me to my eventual research topic. Later, he helped me thread my way through it over three long years. Any virtues in my technical writing are due entirely to him. For his patience and his ever-insightful advice, I sincerely thank Dr. Barry.

I thank Dr. Stuber and Dr. Li for serving on my thesis reading and defense committees. Thanks also to Dr. Lanterman and Dr. Venkateswaran for serving on the defense committee.

In more ways than one, working in Room 562, GCATT taught me how to communicate effectively across noisy channels. For that, and for much else, I thank my labmates. Special thanks to Renato for his insightful comments on my research, on this thesis, and on just about everything else. Andrew set an example with his research, and made one wonder if it was all really as easy as he made it seem. In pitched battles against cold officialdom, Aravind Nayak was often my knight and saviour. Many are the forms I've copied from him, and many the times I've tailed him into some forbidding office, simply saying "Me too" after every one of his sentences. For being a great roommate over so many years, I thank Nayak. Souvik and Shayan held passionate and pointless debates with me on the meaning of life and research. I thank them for their warm friendship and world-class entertainment. Finally, thanks to Arumugam for patiently listening to my discourses on space-time coding. One day, I will stump him with a cricket statistic he doesn't already know. It will be the greatest triumph of my life.

Ravi Sivasankaran was at various times labmate, friend, boss, adversary in debate, political guru, roommate, and fellow enthusiast for Carnatic music. In fact, the only thing he never was is silent. I thank him for talking as much as me, singing much worse than me, and knowing much more than me.

For all the happy times over the years, thanks are due to many friends in Atlanta. Srini Seetharaman taught me to drive, got me out of all the messes I regularly got myself into, and generally parented me through the last two years of my Ph. D. Along with the ever-exuberant Avanti, he is responsible for refining my rather unfortunately colorful wardrobe. Thanks to Shilpa for her constant warmth and patience; to Rajesh for all those grudgingly given dinners; to Amogh for the countless coffee sessions I invited myself to; and to Sridhar for rescuing all my feeble attempts at humor. Nisarga made me feel grown up, which takes some doing.

Sometimes, after an evening shower, blue-gray clouds cover the western sky, and one cannot see the sun. Suddenly, a beam of sunlight breaks through a gap in the clouds, and the newly washed trees glow with a gentle, mellow, golden light. Pradnya, Paka and Lux have often been the human equivalent of that peeping sunbeam. It has been a pleasure and honor knowing and talking to these people. It's a pity that I have nothing worthier than mere words to thank them with.

That I'm actually graduating is due, in large part, to my parents. Every week for the last five years, they anxiously asked me when I'll finish up. At one point, I figured it was easier to graduate than to evade their questions. I thank them for their affection and constant encouragement over the years. They set an example of kindness and simplicity which I'll do well to emulate in the years ahead.

Anthony, Clement and Dilip were proof that greatness exists in real life, outside the verbal abstraction of books. Anthony and Clement taught me much of what I know about myself and about the world. Dilip taught me nothing. He just lived his life. It was teaching enough.

Table of Contents

List of Tables

List of Figures

Summary

The reliability and data rate of wireless communication have traditionally been limited by the presence of multipath fading in wireless channels. However, dramatic performance improvements can be obtained by the use of multiple transmit and receive antennas. Specifically, multiple antennas increase reliability by providing *diversity* gain, namely greater immunity to deep channel fades. They also increase data rates by providing *multiplexing gain*, i.e., the ability to multiplex multiple symbols in one signaling interval.

Harvesting the potential benefits of multiple antennas requires the use of specially designed *space-time* codes at the transmitter front-end. Space-time codes introduce redundancy in the transmitted signal across two dimensions, namely multiple transmit antennas and multiple signaling intervals. In this work, we focus on *linear* space-time codes, which *linearly* combine the real and imaginary parts of their complex inputs to obtain transmit vectors for multiple signaling intervals.

We aim to design optimum linear space-time codes. Optimality metrics and design principles for space-time codes are shown to depend strongly on the codes' function in the overall transmitter architecture. We consider two cases, depending on whether or not the space-time code is complemented by a powerful outer error-control code.

In the absence of an outer code, the multiplexing gain of a space-time code is measured by its rate, while its diversity gain is measured by its raw diversity order. To maximize multiplexing and diversity gains, the space-time code must have maximum possible rate and raw diversity order. We show that there is an infinite set of maximum-rate codes, almost all of which also have maximum raw diversity order. However, different codes in this set have different error rate for a given input alphabet and SNR. Therefore, we develop analytical and numerical optimization techniques to find the code in this set which has the minimum union bound on error rate. Simulation results indicate that optimized codes yield significantly lower error rates than unoptimized codes, at the same data rate and SNR.

In a concatenated architecture, a powerful outer code introduces redundancy in the space-time code inputs, obtaining additional diversity. Thus, the raw diversity order of the space-time inner code is only a lower limit to the total diversity order of the concatenated transmitter. On the other hand, we show that the rate of the space-time code places an upper limit on the multiplexing ability of the concatenated architecture. We conclude that space-time inner codes should have maximum possible rate but need not have high raw diversity order. In particular, the serial-to-parallel converter, which introduces no redundancy at all, is a near-optimum space-time inner code. This claim is supported by simulation results.

On the receiver side, we generalize the well known sphere decoder to develop new detection algorithms for stand-alone space-time codes. These new algorithms are extended to obtain efficient soft-output decoding algorithms for space-time inner codes.

CHAPTER 1

Introduction

Wireless communication systems offer mobility to users and flexibility of deployment to service providers. To gain widespread acceptance, these systems must also be designed to achieve high data rate, while maintaining low error rate. Achieving these twin goals is challenging because wireless channels are not only noisy, but also cause a unique form of distortion known as *multipath fading*. A signal transmitted on the wireless channel gets scattered and reflected from different obstacles in the wireless environment, and hence takes multiple paths to the receiver. At the receiver, these multi-path signals combine constructively or destructively depending on their phase and delay, which in turn are functions of random, time-varying factors like the speed and position of the scatterers, relative to the transmitter and receiver. Consequently, the received signal envelope *fades*, i.e., it varies with time in a random fashion. With a sufficient number of scatterers in the wireless environment, the Rayleigh fading [1] model gives the distribution of the received signal envelope at any given time instant.

One consequence of envelope fading is that the instantaneous signal-to-noise energy ratio (SNR) at the receiver varies randomly with time. Even if the average received SNR is high, the instantaneous received SNR sometimes drops low during a so-called *deep fade*. In a deep fade, errors occur with high probability. One way to reduce the error rate is to

use conventional error correction codes [2]. If these codes are sufficiently long, they can use the relatively noise-free symbols received in periods of high received SNR to recover the symbols lost during a deep fade. However, when fading is slow, deep fades persist for long durations of time, often exceeding the length of one communication packet. In this case, the entire packet is subject to low SNR, and so error correction codes are not very effective. New techniques are required to prevent high error rates due to fading.

1.1 The Benefits of Multiple Antennas

One solution to the problem of deep fades is to use multiple antennas at the transmitter and receiver. The only extra infrastructure required are the antennas themselves, and the power to run the antennas and the supporting RF circuitry. In return, dramatic benefits are obtained.

Suppose there are t transmit and r receive antennas. Counting the number of transmit-receive antenna pairs, we see that there are tr communication links between transmitter and receiver. Further, if the antennas are placed far enough from each other, fading occurs independently on each link. If even one of the tr links is not passing through a deep fade, one can sustain communication between transmitter and receiver, with careful system design. Thus, multiple antennas offer tr-fold fade resistance to deep fades. This is known as the *diversity gain* [3] of multiple antennas, and reflects the fact they can significantly reduce error rates.

In addition to diversity gain, multiple antennas also furnish *multiplexing gain*. More precisely, the transmitter sends t symbols and the receiver obtains r symbols in every signaling interval across the channel. Thus, one can multiplex $\min(t, r)$ information symbols for every channel use, and hence obtain $\min(t, r)$ times the capacity of a single antenna channel [4][5].

In order to harness the potential diversity and multiplexing gains of multiple antennas, the transmitter and receiver should be optimally designed. The novelty in the design problem is that multiple antennas result in a multiple-input, multiple-output (MIMO) fading channel between transmitter and receiver.

We assume that the MIMO channel is a narrowband channel, i.e., signals transmitted in different time instants do not interfere. Further, the receiver knows the MIMO channel, but the transmitter does not. For such a channel, the basic principles of optimum receiver design are well known [6][7]. Therefore, one can adopt a two-step approach to system design for MIMO channels. First, the transmitter is designed assuming the optimum receiver is used. Then, given the transmitter, the known principles of optimum receiver design are implemented. Following this approach, we first discuss transmitter design.

1.2 A Brief Survey of MIMO Transmitter Design Approaches

The diversity and multiplexing gains of multiple antennas reflect their ability to increase the reliability and data-carrying capacity of communication, respectively. The historical evolution of transmitter design for MIMO channels can broadly be split into two distinct paths, depending on whether the primary goal of transmitter design is to increase reliability or capacity. We briefly survey past research in these two directions.

1.2.1 Reliability-Based Approach: Space-Time Codes

The reliability-based approach aims to harness the diversity benefit of MIMO channels to reduce the error rate. In this approach, high data rate is a secondary goal. As mentioned earlier, the diversity gain of MIMO channels comes because there are tr independently fading links between transmitter and receiver. To harvest the diversity gain, each information symbol must be spread across all the available transmit antennas. Conventional forward error correction codes [2] only introduce redundancy across multiple signaling intervals. In a MIMO setting, redundancy must be spread across the multiple transmit antennas and across multiple signaling intervals. Codes which do so are called *space-time* codes [3]. Space-time codes aim to introduce redundancy as cleverly as possible, so as to exploit transmit diversity and reduce error rate.

The cleverness of the redundancy introduced by a space-time code is quantified by its *raw diversity order*, which is the diversity gain obtained when the inputs to the space-time code are independent from one code block to another (i.e., when there is no outer code). Clearly, a high raw diversity order is desirable. It is easy to show [3] that the maximum possible raw diversity order of any space-time code operating over a t-input, r-output Rayleigh fading MIMO channel is tr. This is known as *full* raw diversity order.

To our knowledge, the first codes to obtain full raw diversity order were delay-diversity techniques, which are summarized in [8]. These were later extended to obtain space-time trellis codes [3], which also guarantee full raw diversity order. Arguably the most elegant of all known space-time codes is the Alamouti code [9], which obtains full

raw diversity order, but works only when the number of transmit antennas is two. Subsequently, space-time codes based on *orthogonal designs* [10] generalized the Alamouti code to obtain full raw diversity order for any number of transmit antennas.

While aiming to achieve full raw diversity order, the space-time code should introduce redundancy not only cleverly, but also efficiently. The efficiency is measured by the *rate* of the space-time code, which is defined as the number of information symbols transmitted by the code per signaling interval. Since it is desirable to transmit more information and less redundancy, high rate is desirable. All the codes mentioned above have rate one, except the non-Alamouti orthogonal designs, which have a rate less than one. In some of the early space-time code literature, a rate of one was referred to as full rate, and space-time codes with rate one were considered optimum.

1.2.2 Capacity-Based Approach

In parallel with the development of space-time codes, there was intense information theoretic study of MIMO fading channels. This line of work was pioneered by [4][5], which showed that the data-carrying capacity of a t-input, r-output MIMO channel at high SNR roughly equals $\min(t, r)\log(\text{SNR})$, which is $\min(t, r)$ times the capacity of a scalar channel. Naturally, there was a quest for transmitter architectures which approach this enormous capacity. The V-BLAST architecture [5][11] was the first attempt to do so. The implicit space-time code in a V-BLAST transmitter is a serial-to-parallel (S/P) converter, which takes in t complex symbols every signaling interval and transmits one on each available antenna. Note that there is no attempt to introduce redundancy in order to obtain raw diversity order.

In retrospect, the second major step in the capacity-based approach was the development of *linear dispersion*, or simply, *linear* space-time codes [12]. In each block, these codes take in a finite number of complex input symbols, and use them to generate the transmit signals for a finite number of signaling intervals, called the code length. The defining feature of linear space-time codes is that each output symbol is some linear combination of the inputs and their complex conjugates.

Many reliability-based space-time codes like the Alamouti codes fall under the category of linear codes. However, the primary goal of linear space-time codes, as developed in [12], was not to obtain full raw diversity order, but to achieve the capacity of the MIMO fading channel. It was proved [12][13][14] that linear space-time codes with rate less than $\min(t, r)$ do not achieve the capacity of a t-input, r-output Rayleigh fading channel. Consequently, $\min(t, r)$ is called *full rate*, and capacity-based design aims to develop space-time codes with at least full rate. Note the sharp contrast with the early reliability-based space-time codes, which considered rate one acceptable.

1.3 Unifying the Reliability and Capacity Approaches

It is useful to view the two transmitter design approaches in terms of the metrics they use to measure the goodness of a given space-time code. The reliability approach considers rate and raw diversity order to be valuable, and aims to design space-time codes which maximize both. However, raw diversity order is considered a more valuable asset than rate. Space-time codes with the same rate and raw diversity order do not necessarily have the same error rate, for a given uncoded input alphabet. Thus, one can think of rate and raw diversity order as broad performance metrics. The more precise performance

metric used by the reliability-based approach is the actual error rate, given the SNR and input alphabet to the space-time code. For analytical convenience, the *union bound* on the error rate is also used [3] as a performance metric.

In contrast to the reliability-based approach, the capacity-based approach considers the rate of a space-time code as a more valuable asset than raw diversity order. The inputs to the space-time code are assumed to be coded by an outer error correction code, which is designed to achieve maximum possible data rate for a given SNR. This maximum data rate is then defined to be the *capacity* of a space-time code and is used as the precise performance metric to compare space-time codes.

Naturally, a lot of research has gone into unifying the two approaches to transmitter design. One way to do so is to analyze space-time codes developed from one approach using the metric of the other approach. For example, the capacity of the reliability-based delay diversity techniques was computed in [15]. Similarly, in [14][16], the capacity of the Alamouti code and other orthogonal designs was computed. In particular, it was shown that the Alamouti code for the 2-transmit antenna case loses capacity when there is more than one receive antenna, but achieves maximum possible capacity when there is only one receiver antenna.

The example of the Alamouti code illustrates that it is possible for the same space-time code to be good in both the reliability and capacity senses. Continuing along the same theme, capacity-maximizing linear space-time codes which also have high raw diversity order were found in [17], using approximate numerical techniques. Also, the generalized

layered space-time architecture [18] proposed space-time codes that could trade-off rate and raw diversity order, depending on whether data rate or error rate was deemed more important.

The culmination of the unified design approach was the development of linear complex field (LCF) codes [19][20], also known as threaded algebraic codes [21][22]. LCF codes of length greater than or equal to the number of transmit antennas are guaranteed to achieve full raw diversity order, irrespective of the rate. The only restriction is that the input alphabet to the space-time code should be a lattice alphabet, i.e., the real and imaginary parts of each complex input symbol should be integers.

Of particular interest are LCF codes of rate equal to the number of transmit antennas t, and length greater than or equal to t. These codes are three-way optimal. First, they have rate t greater than or equal to full rate $\min(t, r)$, and are also guaranteed to have full raw diversity order tr. Thus, they satisfy the broad optimality criteria of the reliability and capacity based approaches. Further, they also are optimum with respect to the precise performance metric of the capacity-based approach, i.e., they achieve the capacity of the t-input, r-output MIMO fading channel [20][22]. The only missing link is whether LCF codes are optimum with respect to the precise performance metric of the reliability-based approach, namely whether they achieve minimum word error rate or union bound for a given input alphabet and SNR.

1.4 Contributions of This Work to Transmitter Design

In Chapters 4-6, we show that LCF codes do not achieve the minimum possible union bound, and develop codes that do so. Our approach is as follows. We first show that LCF codes are not the only full-rate, full raw diversity linear codes. In fact, full raw diversity order is surprisingly easy to achieve. Given any rate, we identify an infinite set of linear space-time codes, which also includes LCF codes. With probability one, any randomly picked code from this set achieves full raw diversity order. In particular, space-time codes with rate t and full raw diversity order tr are easy to find. However, the ultimate goal is to not only achieve these two broad properties, but also to optimize the union bound on error rate.

To perform union bound optimization, we first try some analytical techniques. We present a general structure for linear space-time codes that would guarantee the minimum union bound at any SNR and for any input alphabet. However, codes with this structure exist only when either the rate or the raw diversity order (equivalently length) of the space-time code is low. Therefore, our analytical techniques do not solve the union bound optimization problem in general. This forces the use of approximate numerical techniques in order to obtain rate-t, full raw diversity order space-time codes, which also have near-minimum union bound. Simulation results show that these optimized codes achieve significantly lower error rate than LCF codes, for the same data rate and SNR. Further, like the LCF codes, the optimized codes can also be shown to achieve the capacity of the MIMO fading channel. Thus, they are *jointly* optimum with respect to the broad and precise performance metrics of *both* the capacity-based and reliability-based approaches.

In Chapter 7, we take a step back, and investigate whether it is really necessary to use space-time codes which are jointly optimum with respect to both the reliability- and capacity-based approaches. We point out that the two approaches implicitly assume two sharply different transmitter architectures. Given the actual architecture employed, one needs to design space-time codes that are optimum only with respect to the corresponding approach. Our optimized space-time codes, which are jointly optimum with respect to both approaches, are still optimum codes to use in either architecture, but might be a case of overkill. We now describe the two architectures, and why only one approach holds for each of them.

The first architecture contains a *stand-alone* space-time code, whose inputs are independent from one block to the next. Since there is no powerful outer code, this transmitter operates far away from capacity, and so the capacity of the space-time code is not a meaningful measure of actual performance. In contrast, the reliability-based approach is ideally suited to analyze and design such stand-alone space-time codes. In particular, one should aim to achieve full rate and full raw diversity order broadly, and more precisely to obtain minimum possible union bound. The jointly optimum codes we developed earlier are explicitly designed to minimize the union bound, and are hence optimum stand-alone space-time codes. The fact that they also have full outage capacity is unimportant, when one evaluates them as stand-alone space-time codes.

On the other hand, consider a concatenated transmitter architecture where the space-time code acts as an inner code, whose inputs are obtained from a powerful outer code. Even if the space-time inner code has low raw diversity order, the outer code can exploit the full diversity benefit of the channel by coding across different space-time code blocks.

Thus, unlike a good stand-alone space-time code, a good space-time inner code does not need full raw diversity order. This observation implies that the reliability-based approach, with its insistence on high raw diversity order, is not meaningful for designing space-time inner codes.[1] In other words, good space-time inner codes do not need to be good stand-alone space-time codes. In particular, LCF codes and the codes that we obtained by numerically optimizing the union bound might be trying to do too much, in trying to be both good stand-alone *and* good inner space-time codes.

Since the outer code helps the whole concatenated transmitter to approach capacity, the capacity of the space-time code is a meaningful metric, hence the capacity-based approach is ideally suited to analyze space-time inner codes. We proceed to search for optimum space-time inner codes by focusing solely on the capacity.

We define the *multiplexing order* of a space-time code, and use it to show that a rate-R linear space-time code achieves at most a fraction $R/\min(t, r)$ of the capacity of a t-input, r-output Rayleigh fading channel. We also present methods to construct space-time codes that actually achieve this fraction of the channel capacity. The conclusion is that space-time codes with rate less than full rate result in a huge capacity loss, but space-time codes with full rate (or more) can be constructed to avoid this capacity loss.

The crux of the above observations is that good space-time inner codes should have at least full rate, irrespective of their raw diversity order. In particular, a conjecture by Telatar [4] implies that the rate-t S/P converter achieves the capacity of the MIMO fading channel, in spite of its low raw diversity order. Thus, in order to optimize capacity, one can

1. It is well known (see, for example, [17]) that space-time codes that achieve capacity are not necessarily good with uncoded inputs. Our contribution here is only to state that as a fundamental difference between two different transmitter architectures, and to conclude that different design approaches should be adopted for the two architectures.

use the S/P converter instead of the rate-t, full raw diversity order, union-bound optimized codes found in the first half of this work. The advantage of the S/P converter as an inner code is that it is computationally simple to decode. Its disadvantage is that, unlike the more sophisticated optimized codes, it relies on the outer code to provide diversity, and hence does not work well with weak outer codes. However, if the outer code is powerful enough, we conclude that the S/P converter is preferable as an inner code.

1.5 Receiver Design for MIMO Channels

Choosing a particular transmitter architecture fixes the data rate and the structure of the transmitted signal. The task of the receiver is to undo the noisy MIMO channel's distortion, and accurately estimate the transmitted signal. The optimum receiver is the one that minimizes the probability of estimation error. Clearly, the structure of the optimum receiver depends on the coding scheme employed by the transmitter. We now describe the optimum receiver structure for the two transmitter architectures considered in this work. In each case, note that the structure of the optimum receiver is easy to find. The challenge is to develop efficient algorithms to implement the known receiver structure.

1.5.1 Receiver Design for Stand-Alone Space-Time Codes

First, consider the case of a with a stand-alone space-time code, whose inputs are drawn from a discrete input alphabet, independently for each code block. In this case, it is well known [7] that error probability is minimized by using a maximum likelihood (ML) detector at the receiver to estimate the space-time code input in each block. To implement the ML detector, one can think of the combination of a *linear* space-time code and the

MIMO fading channel as a single *effective* MIMO channel, which is also linear. ML detection amounts to a problem of estimating the linear effective channel's input, given its output.

For SISO channels with inter-symbol interference, the Viterbi algorithm implements ML equalization using the trellis as a graphical tool. Similarly, for linear MIMO channels, we develop the class of *tree-pruning* algorithms, which are based on the realization that ML detection amounts to a search for the cheapest leaf node on the *detection tree*. The basic principles of tree-pruning algorithms are simple, and were introduced in [7][23][24]. However, the earlier presentations were developed for different tree-search problems, and had features peculiar to the respective problem. For example, [24] assumes that the code-input alphabet is a possibly infinite lattice. We state the basic principles of tree-pruning algorithms for the specific case of the MIMO detection problem. Also, we point out that these basic principles can be implemented in multiple ways, yielding a wide variety of tree-pruning algorithms. The depth-first tree-search version of the algorithm is identical to the popular sphere decoder [25-30], which is a well-established MIMO detection algorithm. We also develop new tree-pruning algorithms, which require more memory than the sphere decoder, but are more suited to parallel, low-latency implementations.

1.5.2 Receiver Design for Concatenated Transmitters

When the transmitter employs the concatenation of the outer code with an inner space-time code, the optimum receiver must ideally treat the combination of the outer and inner codes as a joint code, and perform ML detection for the joint code. However, this is impractical when the outer code has a large code length. Instead, one follows the turbo

principle [31], and uses an iterative receiver, where soft-output decoders for the inner and outer codes iteratively exchange probabilistic information. In most MIMO transmitters, the outer codes is an algebraic code, for which soft-output decoders are well known [32]. The remaining task is to develop soft-output decoders for the inner space-time code, or equivalently, for the linear effective channel corresponding to the inner space-time code.

Recall that the sphere decoder is a detection algorithm, which produces hard outputs. In [33], the sphere decoder was extended to obtain the *list* sphere decoder, which produces soft outputs by generating a list of hard outputs, instead of just one. We first suggest a new list generation mechanism, that produces the same soft outputs as the method used in [33], but requires lesser computation. Second, we use the list generation mechanism to obtain soft-output extensions of all tree-pruning detection algorithms. Simulation results are shown to illustrate the computational efficiency of our new soft-output algorithms.

1.6 Organization of This Work

A substantial portion of this work, namely Chapters 2-10 focuses on transmitter design. In Chapter 2, we present the channel model and introduce linear space-time codes. The subsequent discussion is divided into the design of stand-alone space-time codes, followed by the design space-time inner codes.

Chapters 3-6 contain our discussion of stand-alone space-time codes. In Chapter 3, we rederive well-known expressions for the union bound, raw diversity order and coding gain of space-time codes. In Chapter 4, we use a random code selection argument to show that full rate, full raw diversity codes are aplenty. In Chapter 5, we explore analytical techniques to optimize the union bound. These techniques are shown to work for some

codes, but not for all. Consequently, in Chapter 6, we develop approximate numerical techniques to optimize full rate full raw diversity codes. We present simulation results to illustrate the benefits of numerical optimization.

Chapter 7 begins our discussion of space-time codes as inner codes by pointing out that the union bound and raw diversity order do not faithfully reflect goodness of inner codes. In Chapter 8, we review information-theoretic analysis of Rayleigh fading channels, focussing in particular on the notion of outage capacity. We introduce the multiplexing order and show that it is equal to $\min(t, r)$. In Chapter 9, we extend the information-theoretic analysis to analyze space-time inner codes. We show that the rate of a space-time code is an upper bound on its multiplexing order, but its raw diversity order is a lower bound on the *achievable* diversity order. This analysis is validated by simulation in Chapter 10, where we also develop broad design rules for space-time inner codes.

Chapters 11 and 12 discuss the design of computationally efficient receivers for MIMO channels. In Chapter 11, we develop new tree-pruning detection algorithms, which are optimum receivers for a transmitter employing a stand-alone space-time code. In Chapter 12, we discuss the iterative receiver structure, which is near-optimum when the transmitter employs a concatenated coding scheme. Specifically, we extend the detection algorithms developed in Chapter 11 to obtain soft-output decoders for linear MIMO channels.

Chapter 13 summarizes the conclusions from this work and discusses areas of future research.

CHAPTER 2

Channel Model and Introduction to Space-Time Codes

An expert is someone who knows more and more about less and less, till
he knows almost everything about almost nothing. — *Unknown*

In this chapter, we describe the multiple-input, multiple-output (MIMO) channel model that will be used in the rest of this work. We motivate space-time codes as a method of harvesting the potential diversity benefit of MIMO channels. The rate and raw diversity order are introduced as two code parameters that determine the data-carrying capacity and error-correcting capability of space-time codes. Finally, we describe the special class of *linear* space-time codes, which will be the focus of the remainder of this work.

2.1 Static-Fading Channel Model for Multiple Antenna Systems

We consider a wireless communication system where the transmitter and receiver are equipped with t and r antennas respectively. In signaling interval k, a $t \times 1$ complex vector $\mathbf{x}_k$ is transmitted across the MIMO wireless channel, yielding an $r \times 1$ complex received vector $\mathbf{y}_k$. In this work, we model the MIMO channel as

- *linear*, i.e., each received signal is a sum of scaled copies of the signals transmitted from all the transmit antennas, and additive white Gaussian noise (AWGN).

- *flat-fading*, or *narrowband*, i.e., there is no interference between symbols transmitted at different signaling intervals. In contrast, frequency-selective or wideband channels have intersymbol interference, and require different design methodologies [6]. Frequency-selective channels will not be considered in this work.

- *quasi-static* or *slow-fading*, i.e., the channel response does not vary over one block of communication. In practice, wireless channels vary with time because of the movement of transmitter, receiver or scatterers in the vicinity. According to the quasi-static assumption, the channel varies so slowly over one communication block that it can be treated as constant. Other possible channel models are the *fast* fading and *block* fading models. The *fast* (or *ergodic*) fading model assumes that the channel varies independently from one signaling interval to the next. The *block fading* assumption is a compromise between the slow and fast extremes. Here, the channel is invariant in small blocks lasting a few signaling intervals, but varies independently from one block to the next. In block fading channels, each communication block, say a codeword, spans multiple channel blocks. These different fading assumptions have different information-theoretic implications [4] and corresponding design methodologies [6]. We restrict ourselves to quasi-static fading in this work.

The input-output relation for a linear, flat-fading, quasi-static MIMO channel is

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k, \tag{1}$$

where the $r \times 1$ noise vector $\mathbf{n}_k$ consists of independent, circular-symmetric, zero-mean complex Gaussian random variables of variance N_0.

The $r \times t$ channel matrix $\mathbf{H}$ is random. We will assume that the transmitter does not know $\mathbf{H}$, but the receiver knows it exactly. This is a crucial and commonly made assumption. If the transmitter knows the channel, pre-coding techniques can be used to simplify system design by converting the MIMO channel to a bank of scalar channels without loss of capacity [6]. However, in most practical systems, the transmitter does not know the channel. In fact, even the receiver often depends on known training symbols from the transmitter in order to estimate the channel. We assume sufficient training, so that the receiver can be assumed to know the channel perfectly. This is realistic except when the channel varies too fast, or when packet-length constraints force short training sequences. Analysis and system design for channels which are unknown to both transmitter and receiver can be found in [6][34][35], but is not considered in this work.

In addition to channel knowledge at transmitter and receiver, another crucial performance and design-determining factor is the distribution of the random channel matrix $\mathbf{H}$. Note that the element h_{ij} of $\mathbf{H}$ denotes the scalar channel between the j^{th} transmit antenna and the i^{th} receive antenna. If the antennas are placed far enough apart, it is both valid and convenient to assume that the coefficients h_{ij} are independent [1]. Further, the *Rayleigh fading* assumption is often made, namely each h_{ij} is assumed to be a complex, zero-mean, circularly symmetric Gaussian random variable of unit variance. For the most part, this work focuses on Rayleigh fading channels. However, some results do hold for more general distributions, as will be explicitly stated where applicable.

Finally, the signal-to-noise energy ratio (SNR) is defined as the ratio of the average received signal energy to the average noise energy, namely

$$S = \frac{\mathrm{E}[\|\mathbf{H}\mathbf{x}_k\|^2]}{rN_0}. \tag{2}$$

2.2 The Benefits of Using Multiple Antennas

With a well-designed communication system, the use of multiple antennas can simultaneously enable higher data rates and lower error rates, at the same SNR.

2.2.1 The Multiplexing Advantage of Multiple Antennas

In every signaling interval across the MIMO fading channel of (1), one can transmit t complex symbols, and receive r complex symbols. Intuitively, one expects the MIMO channel to be able to *multiplex* $\min(t, r)$ symbols in one signaling interval, and consequently carry roughly $\min(t, r)$ [5] times as much data as a single-input, single-output (SISO) channel. This claim will be rigorously proved in Chapter 8 through the discussion of *multiplexing order.*

2.2.2 The Diversity Advantage of Multiple Antennas

The *diversity order* of any communication system measures its reliability at high SNR. When a well-designed communication system transfers data across any channel, the probability of error $P_e(S)$ is typically a decreasing function of the SNR S. The *diversity order* d of the communication system is defined as

$$\delta = -\lim_{S \to \infty} \frac{\log P_e(S)}{\log S}. \tag{3}$$

The diversity order of the *channel* is defined as the diversity order of the best possible communication system that can be used for that channel. The diversity order represents

the asymptotic slope of a log-log plot of error probability vs. SNR. At high SNR, the error probability goes to zero as $\mathrm{SNR}^{-\delta}$, hence a high diversity order is desirable.

For an AWGN channel, error probability can be made to decrease exponentially with SNR using capacity-achieving codes. Consequently, the diversity order of AWGN channels is infinite. In contrast, the diversity order of a t-input, r-output Rayleigh fading channel is

$$\delta(t, r) = tr, \tag{4}$$

as we will see in Chapter 8. The finite diversity order tr places a fundamental limit on the maximum reliability of communication across a MIMO Rayleigh fading channel. Intuitively, fading channels are error-prone because the random channel coefficients occasionally have low energy, even though their average energy is one. Roughly, when a channel coefficient h_{ij} has small energy, we say it is undergoing a *deep fade*. MIMO channels have tr independently fading coefficients $\{h_{ij}\}$ representing tr independent links between transmitter and receiver. Ideally, communication fails only if all links fail. Thus, MIMO channels have a tr-fold resistance to deep fades, as quantified by the diversity order tr in (4).

Harvesting the diversity benefit of multiple receive antennas is conceptually easy. It only requires optimal combining of all available information at the receiver [7]. On the other hand, the diversity benefit of multiple transmit antennas can be obtained only if each information symbol is, in some sense, spread across all available transmit antennas. This was considered conceptually difficult before the introduction [3] of *space-time* codes, which we discuss in the next section.

2.3 Introduction to Space-Time Codes

Space-time codes can be thought of as a transmitter front-end, bridging a single stream of coded or uncoded data with a bank of multiple transmit antennas. As shown in Fig. 1, the input to the transmitter front-end is a serial stream of bits, which are either uncoded, or are obtained from an outer code. The coded/uncoded bits are *modulated* to obtain complex symbols from a finite alphabet, say a QAM or PSK constellation. We will see in later chapters that the presence or absence of the outer code is a crucial factor in determining performance metrics and design criteria for the space-time code.

In one block of encoding, the space-time code takes in a $K \times 1$ complex vector $\mathbf{u}$ containing modulated complex symbols, and produces $t \times 1$ transmit vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ for N signaling intervals. Equivalently, the output of the space-time code is the $t \times N$ complex matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$. The number of signaling intervals per block, namely N, is called the *length* of the space-time code. We will assume that N is finite.

Space-time codes spread information over multiple signaling intervals, just like conventional error correction codes for scalar channels. However, space-time codes also spread information across the multiple transmit antennas. In other words, they introduce redundancy across *space* and *time*. The amount of redundancy introduced by a space-time code is quantified by its *rate*, and the effectiveness of the redundancy is quantified by the *raw diversity order*. We now discuss these two important parameters of a space-time code.

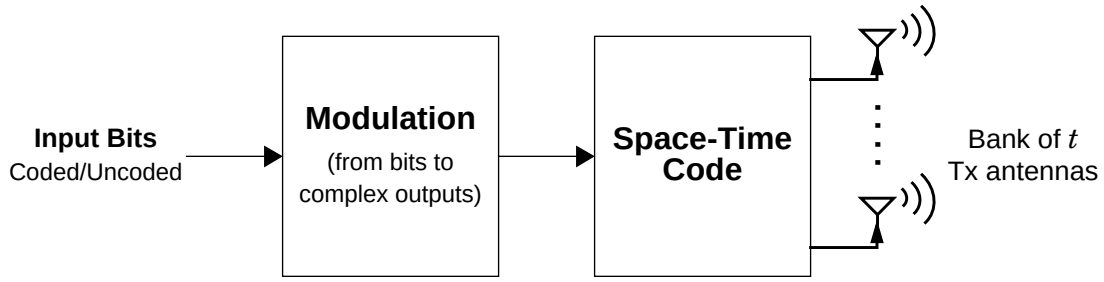

Fig. 1. Block Diagram of transmitter with space-time code as front end.

2.3.1 Rate of a Space-Time Code

The rate of a space-time code is defined as the number of complex input symbols that it encodes per signaling interval. Since we have assumed K inputs in a block lasting N signaling intervals, we get a rate of $R = K/N$. High rate is desirable, because it indicates that a large fraction of the transmitted symbols carry actual information, not redundancy.

Viewed differently, suppose each input symbol to the space-time code is drawn from a QAM (or PSK) constellation of size 2^b. Then, each symbol carries b bits of information. Assuming a pulse shape with zero excess bandwidth, the information rate transmitted by a rate R space-time code with 2^b-QAM input symbols is Rb b / s / Hz. Thus, for the same input constellation, space-time codes with higher rate transmit at a higher data rate. Conversely, to achieve the same data rate, high rate codes can use a smaller constellation.

Another useful view of rate is obtained by considering the effective channel formed by the combination of the space-time code and the MIMO fading channel. The rate is the number of complex inputs multiplexed by the effective channel per signaling interval. Again, high rate implies more multiplexing and is desirable.

2.3.2 Raw Diversity Order of a Space-Time Code

The raw diversity order of a space-time code measures the degree to which it exploits transmit diversity in order to provide fade resistance. In particular, it is the diversity order of a communication system where uncoded bits are fed to the transmitter shown in Fig. 1, and the receiver does optimum maximum-likelihood decoding. Note that by assuming uncoded inputs, the raw diversity order measures the diversity obtained by the space-time code *alone*, without help from an outer code.

In order to compute the raw diversity order of a space-time code, one must, in principle, obtain an expression for the error probability with a given input alphabet, and use it to compute the limit (3). However, the raw diversity order is more conveniently obtained using the *rank* rule, developed in [3]. Given an input alphabet $\mathcal{U}$ containing all possible inputs $\mathbf{u}$ to the space-time code, one obtains the *codebook* $\mathcal{X}$ of all possible $t \times N$ space-time code output matrices $\mathbf{X}$. The rank rule, which we will rederive in Chapter 3, states that the raw diversity order of a space-time code with codebook $\mathcal{X}$ is equal to the product of the number of receive antennas r and the minimum rank of all pairwise differences from the codebook $\mathcal{X}$. In other words,

$$\delta(t, r, \mathcal{X}) = r \min_{\mathbf{X} \neq \mathbf{X}' \in \mathcal{X}} \text{rank}(\mathbf{X} - \mathbf{X}'). \tag{5}$$

Each difference matrix $\mathbf{X} - \mathbf{X}'$ has dimension $t \times N$, and a rank of at most $\min(t, N)$. Thus,

$$\delta(t, r, \mathcal{X}) \leq r\min(t, N), \tag{6}$$

with equality if and only if every pairwise difference between valid code matrices has full rank. In particular, if the code has length $N \geq t$ and full-rank difference matrices, we get the maximum possible, or *full*, raw diversity order, namely tr.

2.4 Linear Space-Time Codes

In this section, we discuss the encoding process of a space-time code. We have said that the output of the space-time code is the $t \times N$ transmit matrix $\mathbf{X}$ whose columns $\mathbf{x}_1$, $\mathbf{x}_2$, …,$\mathbf{x}_N$ are the transmit vectors over the space-time coding block. Equivalently, one can also say that the output is the *composite transmit vector* $\mathbf{x}$ of dimension $Nt \times 1$, given by $\mathbf{x} = [\mathbf{x}_1^{\text{T}}, \mathbf{x}_2^{\text{T}}, \ldots, \mathbf{x}_N^{\text{T}}]^{\text{T}}$. In other words, $\mathbf{x}$ is formed by placing the columns of $\mathbf{X}$ one below

the other. This stacking of columns is denoted $\mathbf{x} = \mathrm{vec}(\mathbf{X})$, and the inverse relation is denoted $\mathbf{X} = \mathrm{mat}(\mathbf{x})$. In general, a space-time code can obtain $\mathbf{x}$ by any operation on the input vector $\mathbf{u}$. However, in this work, we focus on *linear* space-time codes, also known as *linear dispersion codes* [12], where each output symbol is some linear combination of the input symbols and their complex conjugates. In other words, a linear space-time code obtains its composite transmit vector $\mathbf{x}$ according to the rule

$$\mathbf{x} = \mathbf{M}_1\mathbf{u} + \mathbf{M}_2\mathbf{u}^*, \tag{7}$$

where $\mathbf{u}^*$ denotes the complex conjugate of $\mathbf{u}$. $\mathbf{M}_1$ and $\mathbf{M}_2$ are complex $Nt \times K$ matrices. To represent (7) more compactly, we use the complex-to-real transformations [4]

$$\hat{\mathbf{b}} = \begin{bmatrix} \mathrm{Re}\{\mathbf{b}\} \\ \mathrm{Im}\{\mathbf{b}\} \end{bmatrix}, \quad \text{and} \quad \hat{\mathbf{A}} = \begin{bmatrix} \mathrm{Re}\{\mathbf{A}\} & -\mathrm{Im}\{\mathbf{A}\} \\ \mathrm{Im}\{\mathbf{A}\} & \mathrm{Re}\{\mathbf{A}\} \end{bmatrix} \tag{8}$$

for complex vectors $\mathbf{b}$ and matrices $\mathbf{A}$. Now, the encoding rule (7) becomes

$$\hat{\mathbf{x}} = \mathbf{M}\hat{\mathbf{u}}, \tag{9}$$

where the $2Nt \times 2K$ real matrix $\mathbf{M}$ is given by

$$\mathbf{M} = \begin{bmatrix} \mathrm{Re}(\mathbf{M}_1) + \mathrm{Re}(\mathbf{M}_2) & -\mathrm{Im}(\mathbf{M}_1) + \mathrm{Im}(\mathbf{M}_2) \\ \mathrm{Im}(\mathbf{M}_1) + \mathrm{Im}(\mathbf{M}_2) & \mathrm{Re}(\mathbf{M}_1) - \mathrm{Re}(\mathbf{M}_2) \end{bmatrix}. \tag{10}$$

$\mathbf{M}$ is called the *encoding matrix* of the linear ST code, and completely specifies the code.

2.4.1 The Effective Channel for Linear Space-Time Codes

Linear space-time codes also have a simple relation between the code input $\mathbf{u}$ at the transmitter, and the received vectors in the block $\mathbf{y}_1$, $\mathbf{y}_2$, …,$\mathbf{y}_N$. From the MIMO channel equation (1), we have $\mathbf{y}_i = \mathbf{H}\mathbf{x}_i + \mathbf{n}_i$. Now, we assemble the composite receive vector $\mathbf{y} = [\mathbf{y}_1^{\mathrm{T}}, \mathbf{y}_2^{\mathrm{T}}, \ldots, \mathbf{y}_N^{\mathrm{T}}]^{\mathrm{T}}$, and the composite noise vector $\mathbf{n} = [\mathbf{n}_1^{\mathrm{T}}, \mathbf{n}_2^{\mathrm{T}}, \ldots, \mathbf{n}_N^{\mathrm{T}}]^{\mathrm{T}}$. Clearly,

$$\mathbf{y} = \begin{bmatrix} \mathbf{H} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & \ddots & \\ \vdots & \ddots & & \mathbf{0} \\ \mathbf{0} & & \mathbf{0} & \mathbf{H} \end{bmatrix} \mathbf{x} + \mathbf{n} = \mathbf{G}\mathbf{x} + \mathbf{n}, \tag{11}$$

where we have defined the block diagonal transfer matrix $\mathbf{G}$. Now, we only need to apply the complex-to-real transformation (8) to (11) and substitute $\hat{\mathbf{x}} = \mathbf{M}\hat{\mathbf{u}}$ from (9) to obtain

$$\hat{\mathbf{y}} = \hat{\mathbf{G}}\hat{\mathbf{x}} + \hat{\mathbf{n}} = \hat{\mathbf{G}}\mathbf{M}\hat{\mathbf{u}} + \hat{\mathbf{n}}. \tag{12}$$

Equation (12) is the input-output relationship of the effective channel formed by the combination of the linear ST code and the underlying MIMO fading channel, as shown in Fig. 2. From (12), the effective channel is real and has dimensions $2Nr \times 2K$. Remarkably, it is linear and memoryless, just like the underlying MIMO fading channel (1). The linearity of the effective channel makes the analysis and design of linear space-time codes easy, as we will see in chapters Chapter 3 and Chapter 9. The tractability of the effective channel is the primary reason for the general popularity of linear space-time codes.

The linearity of the effective channel also enables the design efficient decoding algorithms for linear space-time codes, as we will see in Chapter 11 and Chapter 12. However, these efficient decoding algorithms work well only when the columns of the effective channel matrix are linearly independent. To intuitively see the need for this restriction, suppose there were no noise $\hat{\mathbf{n}}$. Then, the input $\hat{\mathbf{u}}$ can be obtained from $\hat{\mathbf{y}} =$

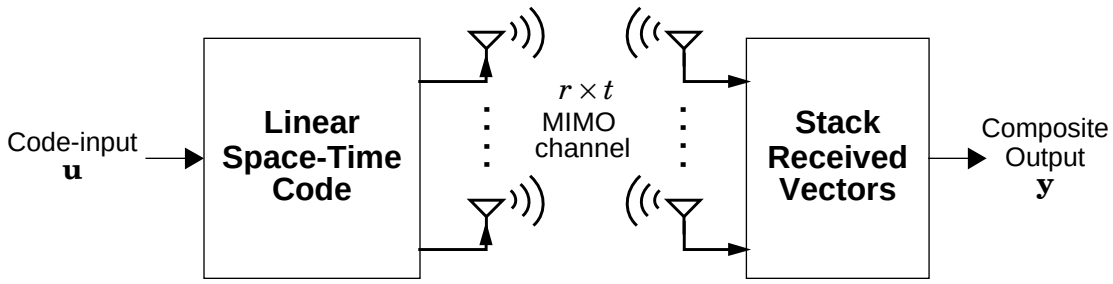

Fig. 2. Effective channel formed by combination of ST code and MIMO fading channel.

$\hat{\mathbf{G}}\mathbf{M}\hat{\mathbf{u}}$ through inversion (or pseudo-inversion) of the effective channel matrix, only if the $2K$ columns of the $\hat{\mathbf{G}}\mathbf{M}$ are linearly independent. Now, the columns are linearly dependent when $\hat{\mathbf{G}}\mathbf{M}$ has rank equal to $2K$. On the other hand, the rank of $\hat{\mathbf{G}}\mathbf{M}$ is at most equal to the rank of $\hat{\mathbf{G}}$, which in turn is at most equal to the minimum dimension of $\hat{\mathbf{G}}$, namely $\min(2Nr, 2Nt)$. Thus, linear independence of columns requires $2K \leq \min(2Nr, 2Nt)$. Dividing both sides by $2N$, we see that the rate R of the linear space-time code should satisfy

$$R = K/N \leq \min(t, r) \tag{13}$$

in order for the efficient decoding techniques to apply.

The value $\min(t, r)$ is called *full rate* of a linear space-time code. From (13), we see that full rate is the maximum rate possible if one also wants efficient decoding. As already mentioned, the rate of a space-time code measures its multiplexing ability. A space-time code operating at full rate multiplexes $\min(t, r)$ symbols per signaling interval, which is equal to the multiplexing ability of the fading channel itself. Rate greater than full rate is possible in theory, but it complicates decoding without increasing the possible multiplexing gain. Consequently, space-time codes are typically designed to have full rate, but not more.

2.4.2 Strictly Linear Space-Time Codes

A sub-class of linear space-time codes is that of *strictly linear* space-time codes, for which the matrix $\mathbf{M}_2$ in (7) is $\mathbf{0}$. The encoding relation is thus

$$\mathbf{x} = \mathbf{M}\mathbf{u}, \tag{14}$$

and the effective channel transfer function is

$$\mathbf{y} = \mathbf{GMu} + \mathbf{n}. \tag{15}$$

For strictly linear space-time codes, the encoding matrix $\mathbf{M}$ and the effective channel matrix $\mathbf{GM}$ are both complex, and have dimensions $Nt \times K$ and $Nr \times K$ respectively. The restriction to strict linearity makes analysis simpler. Heuristically, the best strictly linear space-time codes are found to achieve roughly the same error rate as the best linear space-time codes. However, there is definitely a loss of generality. In particular, no strictly linear space-time code has the elegant properties of the linear Alamouti code (see Section 2.5.2).

2.4.3 Modulation for Linear Space-Time Codes

We close our introduction to linear space-time codes with a remark about the separation of modulation and space-time coding. The modulation process in Fig. 1 just puts a unique, reversible label on the input bits. In general, a space-time code can reverse the label and work with the bits directly if it wants, and so the exact choice of the modulation labels is inconsequential. However, a *linear* space-time code can only perform linear combinations of the modulated complex symbols. Now, the choice of the modulation alphabet affects the codebook of the linear space-time code, and hence the performance. Ideally, one should jointly design the modulation alphabet and encoding matrix together, but the joint design problem is analytically daunting. In this work, we adopt a simpler, sub-optimum alternative [20][40][17], namely we assume QAM or PSK modulation, and then design only the encoding matrix of the linear space-time code.

2.5 Examples of Space-Time Codes

In this section, we present illustrative examples of a few space-time codes. We begin by discussing important linear space-time codes, and end the section by discussing a representative example of non-linear space-time codes.

2.5.1 The Serial-to-Parallel Converter (SPC)

The S/P converter is the simplest space-time code. In one signaling interval, it takes in t complex symbols, and transmits one on each of the t available transmit antennas. Thus, the code length N is 1, the number of inputs K is t, and rate is $K/N = t$. The transmit vector $\mathbf{x}_1$ is equal to the $t \times 1$ input vector $\mathbf{u}$. All non-zero differences between code outputs are non-zero $t \times 1$ vectors, which have rank one. Thus, from the rank rule (5), the S/P converter has a raw diversity order equal to the number of receive antennas r. In spite of its low raw diversity order, the S/P converter is an attractive inner code, when a powerful outer code is present, as we will see in Chapter 9. The V-BLAST transmitter architecture [5][11] employs the S/P converter as inner code.

2.5.2 The Alamouti Code

The Alamouti code [9] assumes that there are $t = 2$ transmit antennas. It takes in the input vector $\mathbf{u} = [u_1\ u_2]^{\mathrm{T}}$ containing $K = 2$ complex symbols, and obtains the code matrix for $N = 2$ signaling intervals, given by

$$\mathbf{X} = \begin{bmatrix} u_1 & -u_2^* \\ u_2 & u_1^* \end{bmatrix}. \tag{16}$$

It is easy to check that if distinct transmit matrices $\mathbf{X}$ and $\mathbf{X}'$ correspond to input vectors $\mathbf{u}$ and $\mathbf{u}'$, then the difference matrix $\mathbf{X} - \mathbf{X}'$ has determinant $\|\mathbf{u} - \mathbf{u}'\|^2$, which is clearly non-zero, implying that all differences $\mathbf{X} - \mathbf{X}'$ are full rank. Thus, from the rank rule (5), the raw diversity order of the Alamouti code is $r\min(t, N) = 2r$, which is equal to the full raw diversity order. On the other hand, since $K = N = 2$, the Alamouti code has a rate $K/N = 1$. This is equal to full rate $\min(t, r)$ if and only if there is exactly $r = 1$ receive antenna.

Beyond rate and raw diversity order, the Alamouti code has one additional feature, namely that the transmit matrix $\mathbf{X}$ always has orthogonal columns, irrespective of the input $\mathbf{u}$. The effect of this property is that the effective channel matrix discussed in Section 2.4.1 always has orthogonal columns, irrespective of the fading channel matrix $\mathbf{H}$. Hence, clever signal processing (see [9]) at the receiver can *diagonalize* the effective channel, giving

$$\mathbf{y}' = \|\mathbf{H}\|_{\mathcal{F}}^2 \mathbf{u} + \mathbf{n}', \tag{17}$$

where $\|\mathbf{H}\|_{\mathcal{F}}^2$ is the squared Frobenius norm (or the energy) in the channel matrix $\mathbf{H}$, and $\mathbf{n}'$ is a noise vector containing independent zero mean complex Gaussian noise terms of variance N_0. The symbols u_1 and u_2 in $\mathbf{u}$ can now be independently decoded, from the corresponding symbols of the effective received vector $\mathbf{y}'$.

Linear space-time codes for which the effective channel can be diagonalized to the form (17) are called *orthogonal designs* [10]. In addition to simple parallel decoding, orthogonal designs also guarantee full raw diversity order. However, it was shown in [10] that all other orthogonal designs have rates less than that of the Alamouti code, namely one. Their low rate is the major drawback of orthogonal designs.

2.5.3 Linear Complex Field Codes

The first full rate, full raw diversity space-time code was developed in [21] for $t = 2$ transmit antennas and $r \geq 2$ receive antennas, using number theoretic ideas. The same ideas were extended to obtain linear complex field codes (LCF) codes (see [19][20][22] and the references therein), that guarantee full raw diversity order for any number of transmit antennas, for all QAM modulated inputs. Though they are often presented as layered or threaded codes, LCF codes are essentially linear space-time codes. Given the number of transmit antennas t and the codelength N, LCF codes exist for any number of code inputs per block satisfying $K \leq Nt$ (or equivalently any rate $R = (K/N) \leq t$). For convenience, we first present the LCF encoding matrix assuming that $K = Lt$, for some integer $L \leq N$. The assumption will be relaxed later. The encoding matrix is parametrized by two unit-magnitude complex numbers α and β. The parameter α determines the $N \times L$ matrix $\mathbf{C}$, according to

$$c_{nl} = \frac{1}{\sqrt{N}} \exp\left(-j\frac{2\pi}{N}(n-1)(l-1)\right)\alpha^{l-1}. \tag{18}$$

The parameter β determines the $t \times t$ diagonal matrix $\mathbf{D}_\beta = \text{diag}(1, \beta, \ldots, \beta^{t-1})$. Let $\mathbf{R}_n$ be the matrix formed by cyclically rotating the rows of the $t \times t$ identity matrix n times, i.e.,

$$\mathbf{R}_n = \begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ \mathbf{I}_{t-n} & \mathbf{0} \end{bmatrix}. \tag{19}$$

With these definitions, the encoding matrix of LCF code parametrized by α and β is

$$\mathbf{M}_{N,L}(\alpha, \beta) = \begin{bmatrix} c_{11}\mathbf{R}_0\mathbf{D}_\beta & c_{12}\mathbf{R}_0\mathbf{D}_\beta & \cdots & c_{1L}\mathbf{R}_0\mathbf{D}_\beta \\ c_{21}\mathbf{R}_1\mathbf{D}_\beta & c_{22}\mathbf{R}_1\mathbf{D}_\beta & \cdots & c_{2L}\mathbf{R}_1\mathbf{D}_\beta \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1}\mathbf{R}_{N-1}\mathbf{D}_\beta & c_{N2}\mathbf{R}_{N-1}\mathbf{D}_\beta & \cdots & c_{NL}\mathbf{R}_{N-1}\mathbf{D}_\beta \end{bmatrix}. \tag{20}$$

The parameters α and β specify the code and determine its codebook, and hence the raw diversity order. It was shown in [19][20][22] that if α is chosen to be an *algebraic integer* [37], the LCF code $\mathbf{M}_{N,L}(\alpha, \alpha^t)$ achieves raw diversity order $\min(t, N)r$ for all QAM input constellations. In particular, LCF codes with $N \geq t$ and $K = Nt$ simultaneously achieve rate t and full raw diversity order. To our knowledge, no other codes with this property have been proposed. However, we will see in Chapter 4 that such codes are aplenty.

Finally, the assumption $K = Lt$ was made only for convenience. Given any $Nt \times Lt$ encoding matrix, one can *puncture* some of the inputs by removing the corresponding columns of the encoding matrix. By puncturing, one can obtain any value of K between 0 and Nt, while still maintaining the raw diversity order of $\min(t, N)r$.

2.5.4 Space-Time Turbo Codes

Space-time turbo codes [38] consist of a binary turbo code, whose output bits are interleaved, modulated, S/P converted, and transmitted on multiple antennas. The interleaver between the outer code and the S/P converter is designed with an additional constraint [38] to ensure full raw diversity order. However, when the turbo code length runs into a few hundred bits, the constraint is loose enough to be ignored, and the interleaver can be arbitrarily chosen without affecting raw diversity order. Then, the combination of the turbo code and the interleaver can be thought of as an outer code, which feeds coded symbols to the inner-space time code, which is only a S/P converter. This separation is convenient because the S/P converter is a linear space-time code, and is therefore more amenable to analysis. We will revisit this separated coding architecture in Chapter 7.

2.5.5 Time-Varying Linear Precoders

Time-varying linear precoders (TVLP), introduced in [15], were designed specifically for *tall* MIMO channels, with more transmit than receive antennas. They are strictly linear space-time codes, except that their encoding matrix varies from block to block, i.e., the matrix $\mathbf{M}_l$ depends on the block index l. Consider the example of length 1 code blocks with one complex input each, with $\mathbf{M}_l = \mathbf{i}_{l \bmod t}$, namely the $(l \bmod t)^{\text{th}}$ column of the $t \times t$ identity matrix. Essentially, every signaling interval, the code takes in one input symbol, and transmits it on just one antenna. The antenna for transmission is chosen in cyclic round-robin fashion. Since only one antenna transmits at any given signaling interval, the effective channel is a time-varying $r \times 1$ channel, which makes decoding simple. However, the simplicity does not come at the cost of diversity. It is easy to show that a well-designed outer code can get full diversity order by decoding soft-outputs from the inner space-time decoder. TVLP codes were probably the first space-time codes to be rigorously analyzed from both diversity and capacity points of view. They have since been overshadowed by linear dispersion codes, particularly because of their dependence on an outer code in order to get diversity. However, their elegance and ease of implementation makes them noteworthy.

2.5.6 Space-Time Trellis Codes

We close this section by discussing space-time trellis codes [3], which are not linear space-time codes. Consequently, the effective channel is not linear, and analysis and decoding of space-time trellis codes tends to be complicated.

Space-time trellis codes are, in one sense, the space-time extension of trellis coded modulation (TCM) for AWGN channels [39]. For every signaling interval k, they maintain a state $\mathbf{s}_k$, containing information about previous inputs and outputs. The state $\mathbf{s}_k$, along with the current input u_k, determines the transmit vector $\mathbf{x}_k$ for that signaling interval. Typically, the state is just a collection of the L previous inputs, namely $\mathbf{s}_k = [u_{k-1}, u_{k-2}, \ldots, u_{k-L}]^{\mathrm{T}}$. L is the called the *memory* of the code. One can represent this encoding process on a trellis, with values of $\mathbf{s}_k$ connected to possible next states $\mathbf{s}_{k+1}$ through *branches*, which are labelled based on the corresponding code input u_k and output vector $\mathbf{x}_k$. The initial state $\mathbf{s}_0$ is pre-determined, and is called the *all-zero* state. After encoding $N - L$ inputs, or equivalently walking across as many trellis stages, the code stops taking inputs, and inserts L termination symbols in order to force the state $\mathbf{s}_N$ to be the all-zero state again. This termination [3] ensures that the raw diversity order of the space-time trellis code is equal to $\min(t, L)r$. By making $L \geq t$, full raw diversity order can be obtained. The next block starts again from the all-zero state, and is independent of the inputs and outputs over the current block.

Note that the output $\mathbf{x}_k$ is not just a linear combination of the inputs $\{u_k\}$ over one block, therefore ST trellis codes are not linear. In fact, the input u_k is just a label for the input bits at time k. Therefore, ST trellis codes can be thought of as taking in bits, and performing joint modulation and coding, as described in Section 2.4.3. Further, the output symbols of a ST trellis code usually belong to a QAM or PSK constellation, thus controlling the peak-to-average power ratio of the transmitted signal. The disadvantage of space-time trellis codes is that when the memory $L \geq t$ is large, the trellis has too many states, making decoding cumbersome.

Table 1: A List of Important Linear Space-Time Codes.

Code	Number of Tx Antennas t	Code Length N	Number of Inputs K	Rate K/N	Raw Diversity Order
S/P converter	Any t	1	t	t	r
Alamouti	2	2	2	1	$2r$
LCF code	Any t	Any N	Any $K \leq Nt$	K/N	$\min(t, N)r$

To our knowledge, space-time trellis codes are the only significant non-linear codes in the literature. There are some algebraic space-time codes [40], which take in bits and produce complex symbols. However, many of these can be split into a combination of a finite-field outer code and a linear inner space-time code. Interestingly, there is one code [41] that actually does independent modulation first, and then does a *non*-linear operation on the modulated complex symbols, without using them merely as labels. This code lacks any generic structure, but can be hand-crafted to outperform linear codes for some channel dimensions and input constellations, according to [41].

The three most important codes for the rest of this work are the S/P converter, the Alamouti code and the LCF code. These are summarized in Table 1, for convenience.

2.6 Importance of Rate and Raw Diversity Order: Example

We now present an example to illustrate that high rate and raw diversity order are crucial in order to achieve good performance. This observation explains the research focus on obtaining full rate, full raw diversity linear space-time codes, both in the literature and in this work. We compare three linear space-time codes operating over a 2-input, 2-output Rayleigh fading channel: the S/P converter, the Alamouti code and a linear complex field

code, whose encoding matrix is given by $\mathbf{M}_{2,2}(\exp(j0.5), \exp(j1.0))$ in (20). In the present example the channel dimensions are $t = r = 2$, hence full rate and full raw diversity order are $\min(t, r) = 2$ and $tr = 4$ respectively. The S/P converter has full rate of $t = 2$, but raw diversity order of only $r = 2$, from Table 1. On the other hand, the Alamouti code has rate one (half of full rate), but full raw diversity order. The LCF code has both full rate and full raw diversity order.

The performance of these codes is compared at 4 bits/s/Hz. In order to achieve this data rate, each input symbol to the S/P converter and the LCF code is drawn independently from a 4-QAM input alphabet. On the other hand, the Alamouti code has half the symbol rate of the other two codes, and consequently has to use a 16-QAM input alphabet. Frames consisting of 100 signaling intervals (corresponding to $100/N = 50$ coded blocks for the length-two Alamouti and LCF codes) are transmitted across the Rayleigh fading channel. The channel is constant over one frame, but varies independently from one frame to the

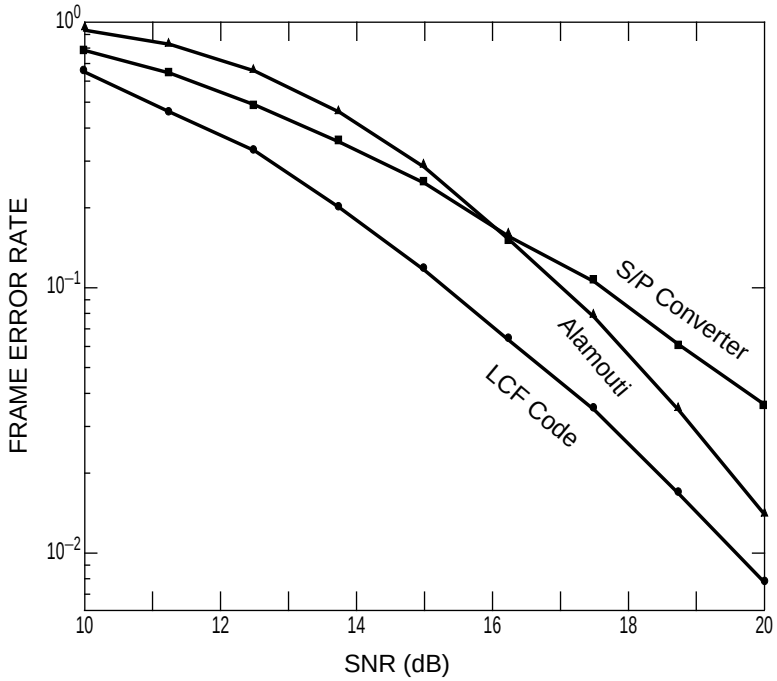

Fig. 3. Performance of three space-time codes over a 2-input, 2-output Rayleigh fading channel at 4 b / s / Hz.

next. The receiver does optimum maximum likelihood decoding, which will be discussed in Section 3.1. A frame error is said to occur if one or more symbols in the frame are decoded erroneously. The resulting frame error rate is plotted against SNR in Fig. 3.

The higher rate of the S/P converter enables it to use a smaller constellation, and hence achieve a lower error rate at low SNR, when compared to the low-rate Alamouti code. However, the full raw diversity order of the Alamouti code leads to a steeper error rate curve than the low diversity S/P converter. Hence, at higher SNR, the Alamouti code outperforms the S/P converter. The LCF code has both full rate and full raw diversity order, hence it outperforms both the other codes.

2.7 Summary

In this chapter, we described the linear, frequency non-selective, slow-fading MIMO channel model (1) used in the remainder of this work. We outlined the multiplexing and diversity benefits of MIMO fading channels, and motivated space-time codes as a transmitter-front end used to harness the available transmit diversity. We defined, and discussed the importance of, the rate and raw diversity order of a space-time code. We defined linear space-time codes. We derived an expression for the encoding process, and for the effective channel formed by the combination of a linear space-time code and the MIMO fading channel. Finally, we presented examples of popular space-time codes.

In the next few chapters, we will assume uncoded inputs to the space-time code, and address the design of linear space-time codes to minimize the error rate, given the channel dimensions, length and rate of the space-time code. Subsequently, the design of space-time inner codes, whose inputs are obtained from an outer code, will be discussed.

CHAPTER 3

Performance Metrics for Stand-Alone Space-Time Codes

In this chapter, we study the performance of space-time codes with uncoded inputs, when the receiver performs optimum maximum likelihood decoding. The system under consideration is shown in Fig. 4. Uncoded input bits are modulated to obtain complex symbols belonging to a countable alphabet. These symbols are encoded by the space-time code to obtain transmit vectors, which are transmitted across a Rayleigh fading channel. The received signals are fed to a maximum likelihood (ML) decoder which estimates the input symbols to the space-time code.

Analysis of the error rate of the communication system of Fig. 4 is part of the standard literature of space-time codes, notably [3][8]. The error rate itself is difficult to obtain in closed form, but the union bound on error rate can be easily derived. Further analysis of the union bound at high SNR yields two new performance metrics, namely the raw diversity order and the coding gain. In subsequent chapters, we will take up each performance metric one at a time, and aim to design space-time codes that optimize it.

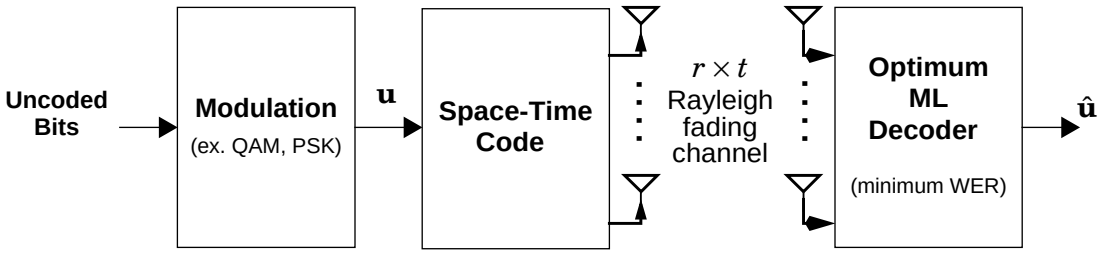

Fig. 4. Transmitter with a stand-alone space-time code and optimum receiver.

3.1 An expression for the Word Error Rate

In this section, we will derive an expression for the error rate of the system shown in Fig. 4. The expression will be shown to be intractable, motivating the derivation of more tractable bounds and approximations.

Recall that each space-time code block lasts N signaling intervals. In one block, the input to the space-time code is a $K \times 1$ complex input vector $\mathbf{u}$, drawn randomly from an input alphabet $\mathcal{U}$. Till the beginning of Chapter 7, we assume that $\mathcal{U}$ is *discrete* or *countable*, i.e., its elements can be indexed by the natural numbers[1]. For example, $\mathcal{U}$ can be the set of all $K \times 1$ lattice vectors, namely vectors whose entries have integer real and imaginary parts. Another example is the smaller set of all $K \times 1$ vectors whose symbols belong to a 16-QAM alphabet. Given the input vector $\mathbf{u}$, the space-time code uses some encoding rule to obtain the $t \times N$ transmit matrix $\mathbf{X}$. The columns of $\mathbf{X}$ are transmitted across the MIMO fading channel over N signaling intervals, and received on r receive antennas. The received vectors constitute the columns of an $r \times N$ receive matrix $\mathbf{Y}$. From the linear memoryless Rayleigh-fading channel model of (1), we get

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}. \tag{21}$$

Given the receive matrix $\mathbf{Y}$ and the channel $\mathbf{H}$, the task of the decoder is to estimate the code input vector $\mathbf{u}$. Whenever the decoder's estimate of the input vector is wrong, a *word error* is said to occur. The probability of word error, called the word error rate (WER), clearly depends on the decoder algorithm used to produce the estimate (see Chapter 11). The decoder that has the minimum WER is the maximum likelihood (ML) decoder. The

1. All finite sets are countable. Not all infinite sets are countable. For example, the infinite set of all rational numbers is countable, while the set of all real numbers is not.

ML decoder uses the conditional probability density function (pdf) $p_{\mathcal{Y}/\mathcal{U}}(\mathbf{Y}\,|\,\mathbf{H},\mathbf{u}')$, which intuitively reflects[1] the probability of receiving the matrix $\mathbf{Y}$ if the actual transmit vector were $\mathbf{u}'$. For AWGN noise with each noise symbol having a variance N_0, the conditional pdf is given by

$$p_{\mathcal{Y}/\mathcal{U}}(\mathbf{Y}/\mathbf{H},\mathbf{u}') = \frac{1}{(\pi N_0)^{tN}} \exp\left(\frac{-\|\mathbf{Y}-\mathbf{H}\mathbf{X}'\|_{\mathcal{F}}^{2}}{N_0}\right), \tag{22}$$

where $\mathbf{X}'$ is the transmit matrix corresponding to the input vector $\mathbf{u}'$.

A vector $\mathbf{u}_1 \in \mathcal{U}$ is said to be *more likely* than another vector $\mathbf{u}_2 \in \mathcal{U}$ if $p_{\mathcal{Y}\,|\,\mathcal{U}}(\mathbf{Y}\,|\,\mathbf{H},\mathbf{u}_1) > p_{\mathcal{Y}/\mathcal{U}}(\mathbf{Y}\,|\,\mathbf{H},\mathbf{u}_2)$. The ML decoder outputs the *most likely* vector in $\mathcal{U}$, namely

$$\hat{\mathbf{u}} = \underset{\mathbf{u}' \in \mathcal{U}}{\arg\max}\; p_{\mathcal{Y}\,|\,\mathcal{U}}(\mathbf{Y}\,|\,\mathbf{H},\mathbf{u}'). \tag{23}$$

The word error rate of the ML decoder is the probability that $\hat{\mathbf{u}}$ is different from the actual input vector $\mathbf{u}$. Note that the word error event depends on three random variables, the actual input vector $\mathbf{u} \in \mathcal{U}$, the random channel matrix $\mathbf{H}$, and the noise matrix $\mathbf{N}$. Let $\mathcal{E}_{\mathbf{u}}$ denote the event that a word error occurs, conditioned on the input vector being one specific $\mathbf{u} \in \mathcal{U}$, and let $\Pr(\mathcal{E}_{\mathbf{u}})$ denote the probability of the event $\mathcal{E}_{\mathbf{u}}$. Averaging $\Pr(\mathcal{E}_{\mathbf{u}})$ over the pdf $p_{\mathcal{U}}(\mathbf{u})$ of the input vector $\mathbf{u}$, we get the word error rate of the ML decoder

$$\mathrm{WER}_{\mathrm{ML}} = \sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \Pr(\mathcal{E}_{\mathbf{u}}). \tag{24}$$

The expression (24) for the word error rate is intractable because $\Pr(\mathcal{E}_{\mathbf{u}})$ cannot easily be computed in closed form. In the next section, we will obtain the union bound on WER by breaking up the conditional word error event $\mathcal{E}_{\mathbf{u}}$ into the union of multiple events.

---

1. The intuitive interpretation is not exact because $\mathbf{Y}$ is a continuous random variable. The pdf directly gives probabilities only for discrete random variables.

3.2 The Union Bound on the Word Error Rate

Let $\mathcal{E}_{\mathbf{u}}(\mathbf{u}')$ denote the *pairwise error event*, that some $\mathbf{u}' \neq \mathbf{u}$ is more likely than the actual transmit vector $\mathbf{u}$. Now, $\mathcal{E}_{\mathbf{u}}$ occurs if any $\mathbf{u}' \neq \mathbf{u}$ is more likely than $\mathbf{u}$, implying

$$\mathcal{E}_{\mathbf{u}} = \bigcup_{\mathbf{u}' \neq \mathbf{u}} \mathcal{E}_{\mathbf{u}}(\mathbf{u}'). \tag{25}$$

Clearly, $\Pr(\mathcal{E}_{\mathbf{u}})$ can be bounded by the sum of *pairwise error probabilities* (PEP), as

$$\Pr(\mathcal{E}_{\mathbf{u}}) \leq \sum_{\mathbf{u}' \neq \mathbf{u}} \Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')). \tag{26}$$

Substituting (26) in (24) gives the *union bound* on the word error rate

$$\mathrm{WER}_{\mathrm{ML}} \leq \sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')). \tag{27}$$

The reason for using the union bound is that the pairwise error probability for each pair ($\mathbf{u}$, $\mathbf{u}'$) is easy to analyze. In particular, a Chernoff bound on the PEP can be obtained by averaging out the random variables determining the pairwise error event, namely the Rayleigh fading channel matrix $\mathbf{H}$ and the noise matrix $\mathbf{N}$. For convenience, the standard derivation [3][8] is reproduced in Appendix A. The final expression is

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) \leq \det{}^{-r}\left(\mathbf{I}_t + \frac{(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^*}{4N_0}\right), \tag{28}$$

where $\mathbf{X}$ and $\mathbf{X}'$ are the transmit matrices corresponding to $\mathbf{u}$ and $\mathbf{u}'$ respectively.

One final manipulation will prove useful to explicitly show the effect of the signal-to-noise energy ratio (SNR) on the union bound. Recall from (2) that the SNR is defined as $S = \mathrm{E}[\|\mathbf{H}\mathbf{x}_k\|^2] / rN_0$. Since each term in the Rayleigh fading channel matrix $\mathbf{H}$ is an independent unit-energy complex Gaussian, it is easy to show that $\mathrm{E}[\|\mathbf{H}\mathbf{x}_k\|^2]/r$ is equal to the average transmit energy $E_{\mathrm{tx}} = \mathrm{E}[\|\mathbf{x}_k\|^2]$. Thus, we can substitute $N_0 = E_{\mathrm{tx}}/S$ in (28),

and substitute, in turn, for the PEP $\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}'))$ into the union bound expression (27). This gives the final expression for the union bound on the word error rate of a space-time code with input alphabet $\mathcal{U}$ and output alphabet $\mathcal{X}$ operating over a t-input, r-output Rayleigh fading channel at SNR S, namely

$$\mathrm{WER}_{\mathrm{ML}} \leq \sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \det{}^{-r}\left(\mathbf{I}_t + S\frac{(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right). \tag{29}$$

An alternative representation of the same union bound will prove useful in the sequel. Each $t \times N$ difference matrix $\mathbf{X} - \mathbf{X}'$ has $\min(t, N)$ ordered singular values, say $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{\min(t,N)} \geq 0$. The number of non-zero singular values is equal to the rank of $\mathbf{X} - \mathbf{X}'$. In terms of the non-zero singular values, the determinant in (29) above can be written as

$$\det\left(\mathbf{I}_t + S\frac{(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right) = \prod_{i=1}^{\mathrm{rank}(\mathbf{X}-\mathbf{X}')}\left(1 + S\frac{\lambda_i^2}{4E_{\mathrm{tx}}}\right). \tag{30}$$

Substituting (30) in (29) yields the equivalent representation of the union bound

$$\mathrm{WER}_{\mathrm{ML}} \leq \sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \prod_{i=1}^{\mathrm{rank}(\mathbf{X}-\mathbf{X}')}\left(1 + S\frac{\lambda_i^2}{4E_{\mathrm{tx}}}\right)^{-r}. \tag{31}$$

We will use (29) and (31) interchangeably, for ease of presentation. Note that the codebook $\mathcal{X}$ of a space-time code determines the pairwise differences, and hence the union bound on WER. Now, we focus on the impact of $\mathcal{X}$ on the WER at high SNR.

3.3 The Raw Diversity Order

Recall from Section 2.2.2 that the diversity order of any communication system is the asymptotic slope of a log error rate vs. log SNR plot. Also, in Section 2.3.2, we briefly introduced the *raw diversity order* of the space-time code with uncoded inputs, i.e., the

raw diversity order of a space-time code is the diversity order of the communication system of Fig. 4. We now proceed to obtain the raw diversity order, using the union bound. By definition, the raw diversity order is the limit

$$\delta(t, r, \mathcal{X}) = -\lim_{S \to \infty} \frac{\log \mathrm{WER}_{\mathrm{ML}}}{\log S}. \tag{32}$$

The union bound expression provides an upper bound on the word error rate, and hence a lower bound on the raw diversity order. It is easy to show that the lower bound is, in fact, tight. So, one can replace the WER in (32) by the union bound, giving

$$\delta(t, r, \mathcal{X}) = -\lim_{S \to \infty} \frac{1}{\log S} \log \sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \prod_{i=1}^{\mathrm{rank}(\mathbf{X}-\mathbf{X}')} \left(1 + S\frac{\lambda_i^2}{4E_{\mathrm{tx}}}\right)^{-r}. \tag{33}$$

At high SNR, the '1 + ' term in each product is negligible, and the SNR exponent in each product is equal to $-\mathrm{rank}(\mathbf{X} - \mathbf{X}')r$. While summing up over all pairs $(\mathbf{u}, \mathbf{u}')$ and averaging over the input probabilities $p_{\mathcal{U}}(\mathbf{u})$, the term with the maximum exponent dominates. Consequently, the raw diversity order is equal to $\rho_{\min} r$, where $\rho_{\min}$ is the minimum rank of pairwise differences $\mathbf{X} - \mathbf{X}'$ between transmit matrices. Thus, we get

$$\delta(t, r, \mathcal{X}) = r \min_{\mathbf{X}' \neq \mathbf{X}} \mathrm{rank}(\mathbf{X} - \mathbf{X}'), \tag{34}$$

as stated in (5) of Section 2.3.2. Since difference matrices have dimension $t \times N$, their maximum possible rank is $\min(t, N)$. Using this fact in (34), we get the rank rule [3][8].

> **Rank Rule:** The maximum raw diversity order of a length N space-time code operating over a t-input, r-output Rayleigh fading channel is equal to $r\min(t, N)$. This upper bound is achieved if and only all pairwise differences $\mathbf{X} - \mathbf{X}'$ between transmit matrices have full rank, equal to $\min(t, N)$.

3.4 The Coding Gain of Space-Time Codes

At high SNR, a plot of log(WER) vs. log(SNR) is a straight line. The raw diversity order is the slope of the asymptote. Two codes with the same raw diversity order achieve the same asymptotic slope, but could differ in the offset (or horizontal shift) of their asymptotes. The *coding gain* of a space-time code is an approximate measure of the offset of the asymptote.

Assume that the input alphabet $\mathcal{U}$ is not only discrete, but also finite. Let $|\mathcal{U}|$ denote the cardinality of $\mathcal{U}$. Then, the inner sum in the union bound (27) can be bounded by

$$\sum_{\mathbf{u}' \neq \mathbf{u}} \Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) \leq (|\mathcal{U}| - 1) \max_{\mathbf{u}' \neq \mathbf{u}} \Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')), \tag{35}$$

just by using the fact that each term $\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}'))$ is less than the maximum, and there are $(|\mathcal{U}| - 1)$ such terms. Now, the union bound is the average of the left hand size over $\mathbf{u}$, and consequently it cannot exceed the maximum of the right hand side over $\mathbf{u}$. This gives

$$\mathrm{WER}_{\mathrm{ML}} \leq \max_{\mathbf{u} \in \mathcal{U}} (|\mathcal{U}| - 1) \max_{\mathbf{u}' \neq \mathbf{u}} \Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')). \tag{36}$$

Intuitively, the right hand side of (36) is obtained when each pairwise error event occurs with the same probability as the *worst case*, or most probable, pairwise error event. Finally, substituting for the PEP $\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}'))$ from (28) in (36), we get the worst-case upper bound on the word error rate,

$$\mathrm{UB}_{\mathrm{WC}} = (|\mathcal{U}| - 1) \max_{\mathbf{u} \in \mathcal{U}} \max_{\mathbf{u}' \neq \mathbf{u}} \prod_{i=1}^{\mathrm{rank}(\mathbf{X} - \mathbf{X}')} \left(1 + S \frac{\lambda_i^2}{4E_{\mathrm{tx}}}\right)^{-r}. \tag{37}$$

At high SNR, the '1 + ' term in the products can be neglected, in comparison with the SNR-dependent term. Also, the terms with the maximum power of SNR, which is by

definition equal to the raw diversity order, will clearly dominate at sufficiently high SNR. Using these high-SNR trends, we get the approximation

$$\mathrm{UB}_{\mathrm{WC}} \approx (|\mathcal{U}| - 1)\, S^{-\delta(t,\, r,\, x)} \max_{\substack{\mathbf{u}' \neq \mathbf{u} \\ \mathrm{rank}(\mathbf{X} - \mathbf{X}') = \rho_{\min}}} \prod_{i\,=\,1}^{\mathrm{rank}(\mathbf{X} - \mathbf{X}')} \left(\frac{\lambda_i^2}{4E_{\mathrm{tx}}} \right)^{-r}. \tag{38}$$

Note that the maximum is taken only over those error events which have the minimum rank difference matrix, since these dominate at high SNR. The reciprocal of this maximum, namely

$$\gamma_{\mathrm{CG}} = \min_{\substack{\mathbf{u}' \neq \mathbf{u} \\ \mathrm{rank}(\mathbf{X} - \mathbf{X}') = \rho_{\min}}} \prod_{i\,=\,1}^{\mathrm{rank}(\mathbf{X} - \mathbf{X}')} \left(\frac{\lambda_i^2}{4E_{\mathrm{tx}}} \right)^{r}, \tag{39}$$

is called the *coding gain* of the space-time code. The coding gain complements the raw diversity order as a measure of code performance at high SNR. The raw diversity order measures the minimum rank, or equivalently, the minimum *number* of non-zero singular values among all difference matrices. From (39), it is easy to see that the coding gain accounts for the actual value of these singular values.

3.5 Comparison of Performance Metrics

In this chapter, we have used bounds and approximations on the word error rate to obtain three performance metrics for a space-time code. The union bound is the most comprehensive of the three performance metrics, and reflects the actual word error rate most closely. One feature of the union bound, obvious from the expression (31), is that it depends on the SNR. At low SNR, it is well known the union bound is a loose bound on

the word error rate, and is therefore not useful. However, as the SNR increases, the union bound becomes tighter. The SNR dependence also makes the union bound somewhat cumbersome as a performance metric, since it needs to be computed afresh for every SNR.

In contrast to the union bound, the raw diversity order and coding gain are SNR-independent, but are meaningful only when the SNR is asymptotically high. Note that neither of them completely describes the word error rate by itself. The raw diversity order gives the slope, and the coding gain attempts to quantify the offset of the word error rate vs. SNR asymptote. Of these, the raw diversity order is clearly the more important. This is because a low-diversity code has a shallower asymptote than a high-diversity code, and has dramatically higher error rates as the SNR increases. In contrast, the effect of a lower coding gain manifests itself in only a shift of the asymptote, and the resultant error rate penalty saturates at high SNR. Further, the coding gain is a pessimistic performance metric, giving often undue importance to the worst-case error event. Thus, it does not exactly measure the offset of the asymptote, but only approximates it.

One special feature of the raw diversity order is that it can be computed easily for any space-time code using the rank rule. This analytical tractability is one of the reasons for its popularity as a design metric.

The three performance metrics derived in this chapter quantitatively measure the goodness of a stand-alone space-time code. Space-time codes with low union bound, or high raw diversity order and coding gain, are desirable. In the next few chapters, we will focus on strictly linear space-time codes, and discuss methods of finding codes which optimize the performance metrics derived here.

CHAPTER 4

Optimization of Raw Diversity Order by Random Code Selection

In Chapter 3, we saw that a space-time code of length N, operating over a t-input, r-output Rayleigh fading channel can obtain at most a raw diversity order of $r\min(t, N)$. In order to achieve this upper bound, the space-time code must satisfy the rank rule. In this chapter, we show that the rank rule is satisfied by almost any strictly linear code whose encoding matrix has orthonormal columns. Thus, the raw diversity order is an easy performance metric to optimize. However, high raw diversity order alone does not guarantee minimum error rate. Consequently, a search for linear space-time codes with minimum union bound is taken up in subsequent chapters.

4.1 The Rank Rule for Strictly Linear Space-Time Codes

In this section, we rederive the rank rule for the specific case of strictly linear space-time codes. Recall from Section 2.4.2 the encoding process of a strictly linear space-time code with encoding matrix $\mathbf{M}$. Given an input vector $\mathbf{u}$ drawn from the countable input alphabet $\mathcal{U}$, the $t \times N$ transmit matrix $\mathbf{X} = \mathrm{mat}(\mathbf{Mu})$ is obtained. The columns of $\mathbf{X}$ are the transmit vectors in one block. The discrete input alphabet $\mathcal{U}$ and the $Nt \times K$ complex encoding matrix $\mathbf{M}$ determine the codebook $\mathcal{X}$ of the space-time code, and hence the performance metrics derived in Chapter 3, specifically the raw diversity order.

From Section 3.3, the raw diversity order of a space-time code is equal to the product of the number of receive antennas and the minimum rank of pairwise differences between valid transmit matrices. In the specific case of strictly linear space-time codes, if $\mathbf{X}$ and $\mathbf{X}'$ are transmit matrices corresponding to two distinct inputs $\mathbf{u}$ and $\mathbf{u}'$, we have

$$\mathbf{X} - \mathbf{X}' = \mathrm{mat}(\mathbf{M}\mathbf{u}) - \mathrm{mat}(\mathbf{M}\mathbf{u}') = \mathrm{mat}(\mathbf{M}(\mathbf{u} - \mathbf{u}')). \tag{40}$$

Let us define the *input difference alphabet* $\mathcal{D}$ as the set of all differences between distinct input vectors, i.e.,

$$\mathcal{D} = \{\mathbf{d} = \mathbf{u} - \mathbf{u}' : \mathbf{u} \neq \mathbf{u}' \in \mathcal{U}\}. \tag{41}$$

If $\mathcal{U}$ is countable, then the difference alphabet $\mathcal{D}$ is also countable.[1] From (41), it is clear that every difference $\mathbf{X} - \mathbf{X}'$ between transmit matrices is equal to $\mathrm{mat}(\mathbf{M}\mathbf{d})$ for some $\mathbf{d} \in \mathcal{D}$. Consequently, the raw diversity order of a strictly linear space-time code with encoding matrix $\mathbf{M}$ and input different alphabet $\mathcal{D}$ operating over a t-input, r-output Rayleigh fading channel is given by

$$\delta(t, r, \mathbf{M}, \mathcal{D}) = r \min_{\mathbf{d} \in \mathcal{D}} \mathrm{rank}(\mathrm{mat}(\mathbf{M}\mathbf{d})). \tag{42}$$

Clearly, the maximum value of the raw diversity order $d(t, r, \mathbf{M}, \mathcal{D})$ is $r\min(t, N)$. According to the rank rule, this maximum raw diversity order is achieved if and only if $\mathrm{mat}(\mathbf{M}\mathbf{d})$ is full rank for all $\mathbf{d} \in \mathcal{D}$. Given t, r and $\mathcal{D}$, optimizing the raw diversity order amounts to choosing an encoding matrix $\mathbf{M}$ so that the rank rule is obeyed. The main result of this chapter is that a randomly chosen $\mathbf{M}$ almost always does the job, as long as it has orthonormal columns. This is stated rigorously in the following theorem.

---

1. The difference alphabet is equivalent to the set of ordered pairs of vectors from the input alphabet. For finite input alphabets, the difference alphabet is also finite, hence countable. For infinite input alphabets, one can invoke Cantor's delightful diagonal trick [44], to prove the countability of the difference alphabet.

Theorem 1. Let a strictly linear space-time code of length N encode $K \times 1$ complex input vectors belonging to a countable input alphabet $\mathcal{U}$, with $K \leq Nt$. Suppose the encoding matrix $\mathbf{M}$ is uniformly picked from the set $\mathcal{M}(Nt, K)$ of all $Nt \times K$ matrices with orthonormal columns. With probability one, the space-time code with encoding matrix $\mathbf{M}$ achieves raw diversity order of $r\min(t, N)$, when operating over a t-input, r-output Rayleigh fading MIMO channel.

The intuition behind the above theorem is as follows. In order to obey the rank rule, the matrix $\mathbf{M}$ has to satisfy a countable number of constraints, namely that $\mathrm{mat}(\mathbf{Md})$ be full rank for each of the countably many difference vectors $\mathbf{d}$. On the other hand, $\mathbf{M}$ is drawn randomly from the *continuous* set $\mathcal{M}(Nt, K)$. The continuity of $\mathcal{M}(Nt, K)$ gives a lot of freedom in the choice of $\mathbf{M}$, making it easy to satisfy the countable number of constraints. As an analogy, note that a random variable drawn uniformly from the interval $[0, 1]$ is almost certainly not equal to $1/n$ for any integer n. This is because the countable set $\{1/n: n = 1, 2, 3, \ldots\}$ is of negligible measure when compared to the continuous set $[0, 1]$. In the remainder of this chapter, we will prove Theorem 1 and discuss its implications.

4.2 The Uniform Distribution on a Continuous Set

In this section, we rigorously discuss what we mean by uniformly picking a random matrix with orthonormal columns, and also offer a practical method to implement this uniform picking. As defined in the theorem statement, let $\mathcal{M}(Nt, K)$ be the set of $Nt \times K$ complex matrices whose columns are orthonormal, namely,

$$\mathcal{M}(Nt, K) = \{\mathbf{M}: \mathbf{M}^*\mathbf{M} = \mathbf{I}_K\}, \tag{43}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix.

$\mathcal{M}(Nt, K)$ is called the *Steifel manifold* [43], and has associated with it a differential *volume element*, $d\mathbf{M}$, and consequently an associated *volume*

$$V(Nt, K) = \int_{\mathcal{M}(Nt, K)} d\mathbf{M}. \tag{44}$$

One can define the *Haar* measure for the volume element $d\mathbf{M}$ in $\mathcal{M}(Nt, K)$ as $p(\mathbf{M})d\mathbf{M}$, where $p(\mathbf{M}) = 1 / V(Nt, K)$ is the probability density function (pdf) corresponding to the Haar measure. Note that the pdf $p(\mathbf{M})$ is the same for all $\mathbf{M} \in \mathcal{M}(Nt, K)$, hence the Haar measure is also called the uniform measure. The Haar measure of any measurable subset $\mathcal{N}$ of $\mathcal{M}(Nt, K)$ is defined as

$$\mu(\mathcal{N}) = \int_{\mathcal{N}} p(\mathbf{M})d\mathbf{M}. \tag{45}$$

We say that a randomly generated matrix $\mathbf{M}$ has been *picked uniformly* from $\mathcal{M}(Nt, K)$, if its pdf is equal to $p(\mathbf{M})$. In practice, such a matrix can be generated by doing Q-L decomposition of a Rayleigh fading matrix of the same dimension. This follows from the following property of Rayleigh fading matrices, which is well known in random matrix theory, and is quite easy to prove [43].

> **Proposition 1.** Consider a random Rayleigh fading matrix $\mathbf{G}$ of dimension $m \times n$, with $m \geq n$. Let $\mathbf{G} = \mathbf{QL}$ be its Q-L[1] decomposition of $\mathbf{G}$. Then, $\mathbf{Q}$ is uniformly distributed over $\mathcal{M}(m, n)$.

Rayleigh fading matrices are easy to generate, using standard methods to generate Gaussian random variables. Thus, Proposition 1 offers a practical method to generate a random matrix $\mathbf{M}$ which is uniformly distributed matrix $\mathcal{M}(Nt, K)$.

---

1. The Q-L decomposition can be done by a Gram Schmidt orthonormalization of the columns of $\mathbf{G}$. One can think of the Q-L decomposition as a transformation of the random matrix $\mathbf{G}$ to the pair ($\mathbf{Q}$, $\mathbf{L}$).

4.3 Proof of Theorem 1

Having defined the uniform distribution, we now proceed to prove Theorem 1. We need to show that if $\mathbf{M}$ is picked uniformly from $\mathcal{M}(Nt, K)$, $\mathrm{mat}(\mathbf{Md})$ has full rank for all $\mathbf{d} \in \mathcal{D}$, with probability one. We first show that for any *one* difference vector $\mathbf{d} \in \mathcal{D}$, $\mathrm{mat}(\mathbf{Md})$ is full rank with probability one, and then extend the result to all the difference vectors using the countability of the alphabet $\mathcal{D}$. The following *rotational invariance* property [43] of uniform distributions over $\mathcal{M}(Nt, K)$ will prove useful.

> **Proposition 2.** Let $\mathbf{M}$ be a random matrix drawn from $\mathcal{M}(Nt, K)$. Then, the distribution of $\mathbf{M}$ is uniform if and only if the transformed matrix $\Theta\mathbf{M}$ has the same distribution as $\mathbf{M}$, for all unitary $Nt \times Nt$ matrices Θ.

Intuitively, the transformation $\Theta\mathbf{M}$ reindexes the elements of the set $\mathcal{M}(Nt, K)$ from which $\mathbf{M}$ is drawn. Proposition 2 says that the uniform distribution is the one and only distribution which *looks alike* at every element of the set $\mathcal{M}(Nt, K)$, and is therefore, invariant to all reindexing of elements.

Of particular interest is the uniform distribution over the set $\mathcal{M}(Nt, 1)$, which contains all $Nt \times 1$ complex vectors with unit Euclidean norm. The following lemmas present two randomly generated vectors that are uniformly distributed over the set $\mathcal{M}(Nt, 1)$.

> **Lemma 1.** For any $\mathbf{d} \neq \mathbf{0}$, if the matrix $\mathbf{M}$ is picked uniformly from $\mathcal{M}(Nt, K)$, the random vector $\mathbf{v} = \mathbf{Md}/\|\mathbf{d}\|$ is uniformly distributed over the set $\mathcal{M}(Nt, 1)$.
>
> **Proof:** Since $\mathbf{M} \in \mathcal{M}(Nt, K)$, it satisfies $\mathbf{M}^*\mathbf{M} = \mathbf{I}_K$. Using this, we obtain $\|\mathbf{Md}\|^2 = \mathbf{d}^*\mathbf{M}^*\mathbf{Md} = \|\mathbf{d}\|^2$. Consequently, the squared norm of $\mathbf{v} = \mathbf{Md}/\|\mathbf{d}\|$ is
>
> $$\|\mathbf{v}\|^2 = \|\mathbf{Md}\|^2/\|\mathbf{d}\|^2 = \|\mathbf{d}\|^2/\|\mathbf{d}\|^2 = 1. \tag{46}$$

Thus, the random vector $\mathbf{v}$ always has unit norm, and so belongs to the set $\mathcal{M}(Nt, 1)$. It remains to prove that it is uniformly distributed on $\mathcal{M}(Nt, 1)$. Let Θ be any unitary $Nt \times Nt$ matrix. Now, from Proposition 2, $\mathbf{M}' = \Theta\mathbf{M}$ has the same distribution as $\mathbf{M}$. Consequently, $\mathbf{v}' = \mathbf{M}'\mathbf{d}/||\mathbf{d}||$ has the same distribution as $\mathbf{v} = \mathbf{M}\mathbf{d}/||\mathbf{d}||$. But clearly $\mathbf{v}' = \Theta\mathbf{v}$. Thus, for any unitary matrix Θ, $\mathbf{v}' = \Theta\mathbf{v}$ has the same distribution as $\mathbf{v}$. Again invoking Proposition 2, we see that $\mathbf{v}$ is uniformly distributed over $\mathcal{M}(Nt, 1)$, which completes the proof.

Lemma 2. Let $\mathbf{G}$ be a Rayleigh distributed random matrix of dimension $t \times N$. Then the vector $\mathbf{w} = \text{vec}(\mathbf{G})/||\mathbf{G}||_{\mathcal{F}}$ is uniformly distributed over the set $\mathcal{M}(Nt, 1)$.

Proof: Let $\mathbf{g} = \text{vec}(\mathbf{G})$ be the $Nt \times 1$ vector formed by stacking the columns of $\mathbf{G}$ one below the other. Since $\mathbf{G}$ is a Rayleigh fading matrix, the entries of $\mathbf{g}$ are independent, unit-variance Gaussian random variables. In this case, it is well known [15] that $\mathbf{g}/||\mathbf{g}||$ is uniformly distributed on $\mathcal{M}(Nt, 1)$. Clearly, $||\mathbf{g}|| = ||\mathbf{G}||_{\mathcal{F}}$. Thus, the uniformly distributed vector $(\mathbf{g}/||\mathbf{g}||)$ equals $\mathbf{w} = \text{vec}(\mathbf{G})/||\mathbf{G}||_{\mathcal{F}}$, completing the proof.

The above two lemmas lead to the proof of the following.

Lemma 3. For any $\mathbf{d} \neq \mathbf{0}$, if the matrix $\mathbf{M}$ is picked uniformly from $\mathcal{M}(Nt, K)$, the $t \times N$ matrix $\text{mat}(\mathbf{Md})$ has full rank $\min(t, N)$ with probability one.

Proof: Consider the random vectors $\mathbf{v} = \mathbf{M}\mathbf{d}/||\mathbf{d}||$ and $\mathbf{w} = \text{vec}(\mathbf{G})/||\mathbf{G}||_{\mathcal{F}}$, where $\mathbf{G}$ is a $t \times N$ Rayleigh fading matrix. From Lemma 1 and Lemma 2, both these vectors are uniformly distributed over $\mathcal{M}(Nt, 1)$ and are therefore equal in distribution. Consequently, the matrices $\text{mat}(\mathbf{v}) = \text{mat}(\mathbf{Md})/||\mathbf{d}||$ and $\text{mat}(\mathbf{w}) = \mathbf{G}/||\mathbf{G}||_{\mathcal{F}}$ are also equal

in distribution. Multiplying by the constant $\|\mathbf{d}\|$, the random matrix $\mathrm{mat}(\mathbf{Md})$ is equal in distribution to $\|\mathbf{d}\|\mathbf{G}/\|\mathbf{G}\|_{\mathcal{F}}$. In particular,

$$\Pr(\mathrm{mat}(\mathbf{Md}) \text{ has full rank}) = \Pr(\|\mathbf{d}\|\mathbf{G}/\|\mathbf{G}\|_{\mathcal{F}} \text{ has full rank}). \tag{47}$$

Now, $\|\mathbf{d}\|$ $(\mathbf{G}/\|\mathbf{G}\|_{\mathcal{F}})$ has full rank if and only if $\mathbf{G}$ itself has full rank. Therefore,

$$\Pr(\mathrm{mat}(\mathbf{Md}) \text{ has full rank}) = \Pr(\mathbf{G} \text{ has full rank}). \tag{48}$$

Rayleigh fading matrices are known [4] to have full rank with probability one. Applying this to the matrix $\mathbf{G}$ and substituting in (48), we see that $\mathrm{mat}(\mathbf{Md})$ has full rank with probability one. This proves the lemma.

Lemma 3 presents the first part of the proof, that a uniformly chosen encoding matrix $\mathbf{M}$ almost certainly satisfies the rank rule for any one difference vector $\mathbf{d}$. Now, we extend this to all the difference vectors and complete the proof of Theorem 1.

Proof of Theorem 1: Let $\mathcal{G}$ denote the event that a random encoding matrix $\mathbf{M}$, uniformly chosen from $\mathcal{M}(Nt, K)$, achieves raw transmit diversity order $\min(t, N)$, and hence raw diversity order $r\min(t, N)$. We want to show that $\mathcal{G}$ occurs with probability one. From the rank rule, $\mathcal{G}$ occurs if and only if $\mathrm{mat}(\mathbf{Md})$ has full rank for all difference vectors $\mathbf{d} \in \mathcal{D}$. Thus,

$$\mathcal{G} = \bigcap_{\mathbf{d} \in \mathcal{D}} \mathcal{G}_{\mathbf{d}}, \tag{49}$$

where $\mathcal{G}_{\mathbf{d}}$ denotes the event that $\mathrm{mat}(\mathbf{Md})$ has full rank. Lemma 3 proves that the probability of $\mathcal{G}_{\mathbf{d}}$ is one. Further, since the input alphabet $\mathcal{U}$ is countable, so is the difference alphabet $\mathcal{D}$. A basic theorem of probability (see, for example [44]) is that the intersection of countably many probability one events also has probability one. Applying this to (49), we see that $\mathcal{G}$ has probability one, completing the proof.

4.4 Implications of Theorem 1: Diversity Is Easy

Theorem 1 holds for any t-input, r-output Rayleigh fading channel, any code length N, and also for any countable input alphabet $\mathcal{U}$. The only restriction is that the number of inputs $K \leq Nt$, since otherwise one cannot find K $Nt \times 1$ orthonormal columns for $\mathbf{M}$. Equivalently, Theorem 1 holds as long as the rate $R = K/N$ of the code is less than or equal to t.

4.4.1 Full-Rate Full-Diversity Codes Are Aplenty

Consider a space-time code with rate equal to full rate $\min(t, r)$ and length $N \geq t$. Since $\min(t, r) \leq t$, the rate restriction is satisfied. Consequently, Theorem 1 applies, implying that almost any code with an orthonormal encoding matrix obtains full raw diversity order of tr. In other words, full rate, full raw diversity codes are aplenty. This is a surprising result, since a high rate amounts to low fractional redundancy, and hence little elbow room for the designer to introduce redundancy cleverly. However, Theorem 1 suggests that maximizing raw diversity order requires almost no cleverness, since almost any random encoding matrix does so. Since no cleverness is required, even the low fractional redundancy implied by full rate is sufficient to guarantee full raw diversity.

4.4.2 Linear Complex Field Codes as a Special Case

The only full rate, full raw diversity codes proposed in the literature are the linear complex field (LCF) codes [20][22], which were discussed in Section 2.5.3. For all rates less than or equal to t, LCF codes guarantee a raw diversity order of $r\min(t, N)$ for lattice-based input alphabets $\mathcal{U}$. LCF codes are also strictly linear codes. Their encoding matrix,

given in (20) of Section 2.5.3 has orthonormal columns, and a special structure based on number-theoretic arguments. Theorem 1 can be viewed as an extension of LCF codes in two ways. First, it holds for *any* countable input alphabet, which is more general than the lattice input alphabet used in LCF codes. Second, it shows that no special number-theoretic structure is required, since almost any encoding matrix with orthonormal columns gets the same (maximum) raw diversity order.

4.5 The Need for Optimizing Other Performance Metrics

The raw diversity of a space-time code determines only the asymptotic slope of a word error vs. SNR curve, but does not completely determine the actual word error rate. Different codes with the same raw diversity order often have significantly different error rates. To illustrate this fact, we present simulation results comparing two complex linear space-time codes operating over a 2-input, 2-output Rayleigh fading channel. Both codes have full rate, namely $R = \min(t, r) = 2$ and length $N = 2$. The first code is an LCF code whose 4×4 encoding matrix is given by $\mathbf{M}_{2,2}(\exp(j0.5), \exp(j1.0))$ in (20). For the second code, the encoding matrix was randomly generated from the set $\mathcal{M}(4, 4)$ of all 4×4 matrices with orthonormal columns using the method of Proposition 1.

In each space-time code block, the $K = NR = 4$ input symbols to both space-time codes are randomly and independently drawn from a 4-QAM constellation. Since two such symbols are transmitted per signaling interval, the data rate is 4 b / s / Hz. Frames consisting of 50 coded blocks (or equivalently $50N = 100$ signaling intervals) are transmitted across the Rayleigh fading channel.[1] The Rayleigh fading channel is constant

1. For the second code, the *same* random encoding matrix was used for all frames.

over one frame, but varies independently from one from to the next. The receiver performs optimum ML decoding using a sphere decoder (see [25] and Chapter 11). A frame error is said to occur if one or more symbols in the frame are decoded erroneously.

The resulting frame error rate is plotted against SNR for the two codes in Fig. 5. The LCF code is guaranteed to achieve full raw diversity order, namely $tr = 4$. Theorem 1 predicts that with probability one, the random code should also achieve full raw diversity order. Fig. 5 confirms this prediction, since the curves for both codes are visually seen to have the same asymptotic slope. However, the random code does not achieve the same frame error rate as the LCF code. It needs about 0.5 dB more SNR than the LCF code to achieve the same frame error rate. This observation leaves open the possibility that the error rate can be reduced below that of the LCF code using another optimized complex linear space-time code.

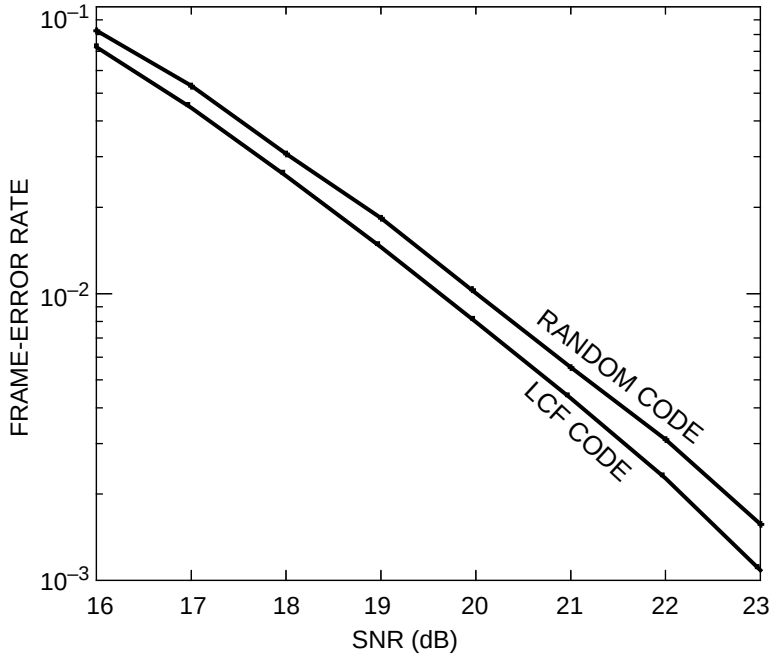

Fig. 5. Performance of LCF and random linear space-time codes over a 2-input, 2-output Rayleigh fading channel at 4 bits / s / Hz.

4.6 Summary

In this chapter, we have shown that all encoding matrices with orthonormal columns have the same (optimum) raw diversity order. However, simulation results show that the raw diversity order alone does not guarantee that the minimum possible error rate is achieved. In other words, high raw diversity order is a loose performance metric: it is necessary but not sufficient to guarantee low error rates. The looseness of the raw diversity order motivates us to look at the other performance metrics derived in Chapter 3, namely the union bound and coding gain. This will be the topic of the next two chapters.

The raw diversity order is an integer-valued function of the continuous-valued encoding matrix, and is consequently easy to optimize. On the other hand, we will see in the next chapter that optimizing the continuous-valued union bound is a more daunting problem.

CHAPTER 5

Analytical Results on Optimization of the Union Bound

In this chapter, we aim to find strictly linear space-time codes that minimize the union bound on word error rate. Most of the discussion will be focussed on the specific case where the encoding matrix has orthonormal columns. Our goal is to analytically find an orthonormal-column encoding matrix that achieves the minimum union bound among all such matrices. We will derive the *orthogonal differences* bound, and use it to obtain optimum encoding matrices of some specific dimensions. Unfortunately, for all other matrix dimensions, we will see that the bound cannot be used solve the union bound optimization problem. In particular, the bound is not useful to optimize full rate, full raw diversity order codes, forcing us to resort to numerical optimization in the next chapter.

5.1 The Union Bound for Strictly Linear Space-Time Codes

We begin this section by rederiving the union bound specifically for strictly linear space-time codes. Recall from (29) the union bound for a general space-time code

$$\sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \det{}^{-r}\left(\mathbf{I}_t + S\frac{(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right). \tag{50}$$

For a strictly linear space-time code we already saw in (40) that $\mathbf{X} - \mathbf{X}' = \mathrm{mat}(\mathbf{M}(\mathbf{u} - \mathbf{u}'))$. Thus, the union bound (50) becomes

$$\sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \det{}^{-r}\left(\mathbf{I}_t + S\frac{\mathrm{mat}(\mathbf{Md})(\mathrm{mat}(\mathbf{Md}))^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right), \tag{51}$$

where $\mathbf{d} = \mathbf{u} - \mathbf{u}'$. Assembling all the terms with the same difference vector $\mathbf{d} = \mathbf{u} - \mathbf{u}'$ together, we obtain a simpler expression for the union bound, namely

$$P_{\mathrm{UB}}(S, \mathcal{D}, \mathbf{M}) = \sum_{\mathbf{d} \in \mathcal{D}} p_{\mathcal{D}}(\mathbf{d}) \det{}^{-r}\left(\mathbf{I}_t + S\frac{\mathrm{mat}(\mathbf{Md})(\mathrm{mat}(\mathbf{Md}))^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right), \tag{52}$$

where $\mathcal{D} = \{\mathbf{d} = \mathbf{u} - \mathbf{u}' : \mathbf{u} \neq \mathbf{u}' \in \mathcal{U}\}$ is the input difference alphabet familiar from (41). Equivalently, writing each determinant in terms of the singular values of $\mathrm{mat}(\mathbf{Md})$, we get

$$P_{\mathrm{UB}}(S, \mathcal{D}, \mathbf{M}) = \sum_{\mathbf{d} \in \mathcal{D}} p_{\mathcal{D}}(\mathbf{d}) \prod_{i=1}^{\mathrm{rank}(\mathbf{Md})} \left(1 + S\frac{\lambda_i^2}{4E_{\mathrm{tx}}}\right)^{-r}. \tag{53}$$

Note that the union bound above depends on the SNR, the input difference alphabet and the $Nt \times K$ encoding matrix $\mathbf{M}$[1]. In this chapter, we will focus specifically on encoding matrices with orthonormal columns, i.e., matrices belonging to the set $\mathcal{M}(Nt, K) = \{\mathbf{M} : \mathbf{M}^{\mathrm{H}}\mathbf{M} = \mathbf{I}_K\}$. From the previous chapter, almost all matrices in $\mathcal{M}(Nt, K)$ have the optimum raw diversity order, namely $r\min(t, N)$. However, we will see that the matrices differ significantly in the union bound. Our goal is to find an optimum matrix $\mathbf{M} \in \mathcal{M}(Nt, K)$, i.e., one that has the minimum possible value of $P_{\mathrm{UB}}(S, \mathcal{D}, \mathbf{M})$.

An alternative representation of the encoding matrix $\mathbf{M}$ will prove useful for the analysis. Let the orthonormal columns of $\mathbf{M}$ be the vectors $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K$. The i^{th} *folded column* of $\mathbf{M}$ is defined to be the $t \times N$ matrix $\mathbf{M}_i$ obtained by reshaping the i^{th} column $\mathbf{m}_i$.

1. For ease of notation, we do not explicitly include the channel dimensions as parameters.

In other words, $\mathbf{M}_i = \mathrm{mat}(\mathbf{m}_i)$. Clearly, there is a unique correspondence between the folded columns $\{\mathbf{M}_i\}$ and the encoding matrix $\mathbf{M}$. In particular, $\mathbf{M} \in \mathcal{M}(Nt, K)$ if and only if its columns are orthonormal, or equivalently, if and only if the folded columns satisfy

$$\mathrm{tr}(\mathbf{M}_i^* \mathbf{M}_j) = \mathbf{m}_i^* \mathbf{m}_j = \delta_{ij}. \tag{54}$$

Finally, since $\mathbf{Md} = \sum_{i=1}^{K} \mathbf{m}_i d_i$, we see that

$$\mathrm{mat}(\mathbf{Md}) = \sum_{i=1}^{K} \mathrm{mat}(\mathbf{m}_i) d_i = \sum_{i=1}^{K} \mathbf{M}_i d_i. \tag{55}$$

Substituting for $\mathrm{mat}(\mathbf{Md})$ from (55) in (53), we get the union bound as a function $P_{\mathrm{UB}}(S, \mathcal{D}, \{\mathbf{M}_i\})$ of the folded columns of $\mathbf{M}$. Thus, the optimization problem is to find K matrices $\{\mathbf{M}_i\}$ of dimension $t \times N$ satisfying (54) that minimize the union bound (53).

5.2 There Are Infinitely Many Optimum Encoding Matrices

In this section, we will show that there are infinitely many choices of $\{\mathbf{M}_i\}$ that minimize the union bound. First, note that the union bound (53) is a bounded, continuous function of $\{\mathbf{M}_i\}$. Also, the set of all matrices $\{\mathbf{M}_i\}$ satisfying (54) is bounded and closed. It is a well-known result in function theory that a bounded, continuous function has a point of minimum in any closed, bounded set. Consequently, there is at least one optimum choice of the folded columns $\{\mathbf{M}_i\}$. The following remark shows that this optimum choice is not unique. (A slightly different version of the same result can be found in [12][17].)

Remark 1. Suppose two sets of folded columns $\{\mathbf{M}_i\}$ and $\{\mathbf{M}_i'\}$ are related by

$$\mathbf{M}_i' = \mathbf{Q}_l \mathbf{M}_i \mathbf{Q}_r, \tag{56}$$

where $\mathbf{Q}_l$ and $\mathbf{Q}_r$ are some two unitary matrices of dimensions $t \times t$ and $N \times N$ respectively. Then, these two sets have the same union bound (53), irrespective of $\mathcal{D}$.

Proof: For any difference vector $\mathbf{d}$, we see from (55) that

$$\mathrm{mat}(\mathbf{M}'\mathbf{d}) = \sum_{i=1}^{K} \mathbf{M}_i' d_i, \text{ and } \mathrm{mat}(\mathbf{M}\mathbf{d}) = \sum_{i=1}^{K} \mathbf{M}_i d_i. \tag{57}$$

From the above, the assumed relation (56) between $\{\mathbf{M}_i\}$ and $\{\mathbf{M}_i'\}$ yields

$$\mathrm{mat}(\mathbf{M}'\mathbf{d}) = \mathbf{Q}_l \mathrm{mat}(\mathbf{M}\mathbf{d})\mathbf{Q}_r. \tag{58}$$

By assumption, the matrices $\mathbf{Q}_l$ and $\mathbf{Q}_r$ are unitary rotation matrices. Multiplication by them does not change the singular values of a matrix. Consequently, the singular values of $\mathrm{mat}(\mathbf{M}'\mathbf{d})$ and $\mathrm{mat}(\mathbf{M}\mathbf{d})$ are identical for all $\mathbf{d}$. Since the union bound (53) depends only these singular values, $\mathbf{M}'$ and $\mathbf{M}$ have the same union bound.

Given any encoding matrix $\mathbf{M}$, one can obtain infinitely many encoding matrices $\mathbf{M}'$ with the same union bound, simply by choosing $\mathbf{Q}_l$ and $\mathbf{Q}_r$ and transforming the columns as shown in (56). Since at least one optimum encoding matrix is guaranteed to exist, we can find infinite other optimum encoding matrices, using this column transformation.

5.3 The Orthogonal Differences (OD) Bound

We now derive an expression for the least union bound that can be achieved by any encoding matrix. Later, we will use this expression to solve the union bound optimization problem. Consider the determinant in each term constituting the union bound, namely

$$\det\left(\mathbf{I}_t + \frac{S}{4E_{\mathrm{tx}}}\mathrm{mat}(\mathbf{M}\mathbf{d})(\mathrm{mat}(\mathbf{M}\mathbf{d}))^{\mathrm{H}}\right) = \prod_{i=1}^{\min(t,N)} \left(1 + \frac{S}{4E_{\mathrm{tx}}}\lambda_i^2\right). \tag{59}$$

Now, the sum of the squared singular values of any matrix is equal to the energy in that matrix. Thus, the singular values $\{\lambda_i\}$ of the matrix $\mathrm{mat}(\mathbf{Md})$ satisfy the sum constraint

$$\sum_{i=1}^{\min(t,N)} \lambda_i^2 = \|\mathrm{mat}(\mathbf{Md})\|_{\mathcal{F}}^2 = \|\mathbf{Md}\|^2. \tag{60}$$

Under the sum constraint (60), it is easy to show[1] that the determinant (59) is maximized if and only if each λ_i^2 equals $\|\mathbf{Md}\|^2/\min(t, N)$. Using this fact, we get the bound

$$\det\left(\mathbf{I}_t + \frac{S}{4E_{\mathrm{tx}}}\mathrm{mat}(\mathbf{Md})(\mathrm{mat}(\mathbf{Md}))^{\mathrm{H}}\right) \le \left(1 + \frac{S\|\mathbf{Md}\|^2}{4E_{\mathrm{tx}}\min(t,N)}\right)^{\min(t,N)}. \tag{61}$$

Substituting the above in (52) gives the *orthogonal differences* (OD) bound

$$P_{\mathrm{UB}}(S, \mathcal{D}, \mathbf{M}) \ge \sum_{\mathbf{d}\in\mathcal{D}} p_{\mathcal{D}}(\mathbf{d})\left(1 + \frac{S\|\mathbf{Md}\|^2}{4E_{\mathrm{tx}}\min(t,N)}\right)^{-r\min(t,N)}. \tag{62}$$

One problem with the OD bound is the term $\|\mathbf{Md}\|$ on the right hand side, which in general depends on the encoding matrix $\mathbf{M}$. Consequently, the OD bound (62) merely bounds $P_{\mathrm{UB}}(S, \mathcal{D}, \mathbf{M})$ by another function of $\mathbf{M}$. This is not very useful because the new bounding function is almost as intractable as the original function itself. However, for encoding matrices with orthonormal columns, the OD bound is particularly useful. To see this, note that $\|\mathbf{Md}\| = \|\mathbf{d}\|$ for all $\mathbf{M} \in \mathcal{M}(Nt, K)$. Consequently, the terms $\|\mathbf{Md}\|$ become independent of $\mathbf{M}$, hence the OD bound gives a benchmark against which all matrices in $\mathcal{M}(Nt, K)$ can be compared. Also, note that the OD bound is achieved if and only if the $\min(t, N)$ singular values of $\mathrm{mat}(\mathbf{Md})$ are equal, for all difference vectors $\mathbf{d} \in \mathcal{D}$. Equivalently, either

1. The sum constraint (60) fixes the arithmetic mean of the terms in the terms constituting the product (59). Based on this observation, one can prove (61) using the geometric inequality, which says that the geometric mean of a collection of terms is at most equal to the arithmetic mean, with equality if and only if all the terms are equal.

the rows or columns of $\mathrm{mat}(\mathbf{Md})$, whichever are fewer in number, should be orthogonal. Reverting to the folded columns representation of the encoding matrix, we sum up the OD bound below.

Remark 2. The union bound (53) of any set of K folded columns $\{\mathbf{M}_i\}$ satisfying (54) satisfies the orthogonal differences bound

$$P_{\mathrm{UB}}(S, \mathcal{D}, \{\mathbf{M}_i\}) \geq \sum_{\mathbf{d} \in \mathcal{D}} p_{\mathcal{D}}(\mathbf{d}) \left(1 + \frac{S\|\mathbf{d}\|^2}{4E_{\mathrm{tx}} \min(t, N)}\right)^{-r \min(t, N)} \tag{63}$$

with equality if and only if the $t \times N$ matrix $\mathrm{mat}(\mathbf{Md})$ has equal singular values for all $\mathbf{d} \in \mathcal{D}$. Equivalently, representing conjugate transpose by *, $\mathrm{mat}(\mathbf{Md})$ must satisfy

$$\mathrm{mat}(\mathbf{Md})^* \mathrm{mat}(\mathbf{Md}) = (\|\mathbf{d}\|^2/N)\, \mathbf{I}_N, \quad \text{if } t \geq N, \tag{64}$$

$$\mathrm{mat}(\mathbf{Md}) \mathrm{mat}(\mathbf{Md})^* = (\|\mathbf{d}\|^2/t)\, \mathbf{I}_t, \quad \text{if } t < N. \tag{65}$$

The OD bound above can be thought of as a sufficient condition for optimality. If a set of K folded columns $\{\mathbf{M}_i\}$ achieves the OD bound (63) with equality, it is guaranteed to be optimal, since no other choice can do better. We will find such optimum sets in the next section for some values of t, N and K. On the other hand, the OD bound is not a necessary condition for optimality. In other words, there is no guarantee that there is some set of K folded columns $\{\mathbf{M}_i\}$ that actually achieves the OD bound with equality. In fact, for a wide range of t, N and K, we will see in a subsequent section that the OD bound is in fact unreachable.

5.4 Solution to Union Bound Optimization for Some Special Cases

In this section, we will construct optimum encoding matrices for space-time codes whose parameters satisfy $K \le \max(t/N, N/t)$. The crucial step is the following sufficient condition for the folded columns $\{\mathbf{M}_i\}$ in order to achieve the OD bound.

Lemma 4. Any set of K folded columns $\{\mathbf{M}_i\}$ that satisfies

$$\mathbf{M}_i^* \mathbf{M}_j = \frac{1}{N} \delta_{ij} \mathbf{I}_N \text{ if } t \ge N, \qquad \mathbf{M}_i \mathbf{M}_j^* = \frac{1}{t} \delta_{ij} \mathbf{I}_t \text{ if } t < N \tag{66}$$

achieves the OD bound, irrespective of the input difference alphabet $\mathcal{D}$.

Proof: For convenience, we will prove this assuming $t \ge N$. In this case, from (64), the OD bound is achieved if and only if $\mathrm{mat}(\mathbf{Md})^* \mathrm{mat}(\mathbf{Md}) = (\|\mathbf{d}\|^2/N)\, \mathbf{I}_N$. We need to show that (66) suffices to ensure this. This follows from straightforward substitution. Note that since $\mathrm{mat}(\mathbf{Md}) = \sum_{i=1}^{K} \mathbf{M}_i d_i$, we have

$$\mathrm{mat}(\mathbf{Md})^* \mathrm{mat}(\mathbf{Md}) = \sum_{i=1}^{K} \sum_{j=1}^{K} d_i^* d_j \mathbf{M}_i^* \mathbf{M}_j. \tag{67}$$

Substituting (66) in (67), we get

$$\mathrm{mat}(\mathbf{Md})^* \mathrm{mat}(\mathbf{Md}) = \sum_{i=1}^{K} \sum_{j=1}^{K} d_i^* d_j \frac{1}{N} \delta_{ij} \mathbf{I}_N = (\|\mathbf{d}\|^2/N)\, \mathbf{I}_N, \tag{68}$$

which is the required condition in (64). This proves the lemma.

For convenience, we will say that

Definition 1. The folded columns $\{\mathbf{M}_i\}$ are *strongly orthogonal* if they satisfy (66).

Note that the strong orthogonality of (66) is a stronger restriction on the folded columns $\{\mathbf{M}_i\}$ than (54), which is imposed by the orthogonality of the encoding matrix. Condition (54) only requires that when $i \neq j$, the matrix $\mathbf{M}_i^* \mathbf{M}_j$ have *trace* zero. On the other hand, strongly orthogonality requires that all the elements of $\mathbf{M}_i^* \mathbf{M}_j$ be identically zero, when $t \geq N$.

Lemma 4 says that in order to achieve the OD bound, it is sufficient to find K folded columns $\{\mathbf{M}_i\}$ of dimension $t \times N$ that are strongly orthogonal. Now, we will construct such folded columns, assuming $K \leq \max(t/N, N/t)$. Take the case $t \geq N$, and consider the collection of all the $t \times 1$ columns of the matrices $\{\mathbf{M}_i\}$. Clearly, the condition $\mathbf{M}_i^* \mathbf{M}_j = (\mathbf{I}_N / N)\, \delta_{ij}$ in (66) holds if and only if that all these columns are mutually orthogonal, and each column has norm $1/N$. Now, a simple counting argument can be used. The number of required orthogonal columns is NK, since there are K matrices with N columns each. Now, NK orthogonal columns of dimension $t \times 1$ exist if and only if $NK \leq t$, or $K \leq (t/N)$. In the case $t < N$, the same argument applies, except that we will interested in making all the Nt rows of $\{\mathbf{M}_i\}$ mutually orthogonal. Again, that is possible if and only if $K \leq (N/t)$. The following proposition formally combines these two statements.

Proposition 3. Given (t, N, K), one can find a set of K strongly orthogonal folded columns $\{\mathbf{M}_i\}$ of dimension $t \times N$, if and only if

$$K \leq \max\left(\frac{t}{N}, \frac{N}{t}\right). \tag{69}$$

Suppose the condition (69) is satisfied. If $t \geq N$, any set of NK orthogonal $t \times 1$ vectors can be used as the columns of $\{\mathbf{M}_i\}$. Conversely, if $t < N$, any set of tK orthogonal $1 \times N$ vectors can be used as the rows of $\{\mathbf{M}_i\}$.

Proposition 3 gives a constructive method to obtain strongly orthogonal folded columns, which are then guaranteed to achieve to achieve the minimum possible union bound, according to Lemma 4. In the following, we will give examples of optimum encoding matrices found using Proposition 3.

5.4.1 Optimum Encoding Matrices for $K = 1$ input

We first consider the case where the number of space-time code inputs is $K = 1$. In this case, the encoding matrix is just an $Nt \times 1$ vector $\mathbf{m}$ of norm one, and there is just one folded column, namely $\mathrm{mat}(\mathbf{m})$. Note that $K = 1$ satisfies the condition $K \leq \max(t/N, N/t)$ for all t and N. From Proposition 3, $\mathrm{mat}(\mathbf{m})$ can be chosen to achieve the OD bound. In particular, the following remark presents the form of $\mathbf{m}$ that minimizes the union bound.

> **Remark 3.** Let $\mathbf{m}$ be the encoding matrix of a strictly linear space-time code encoding one complex input in N signaling intervals across t-input, r-output Rayleigh fading channel. Then, the union bound on word error rate is minimum if $\mathrm{mat}(\mathbf{m})$ has orthogonal columns if $t \geq N$, and orthogonal rows if $t < N$.

An interesting observation relates to the encoding matrix of the linear complex field (LCF) codes, given in (20) of Section 2.5.3. By inspection, it is easy to see that each folded column of the encoding matrix has orthogonal rows or columns, implying that LCF codes are optimum for the single-input case. This is stated in the following.

> **Corollary 1.** When the number of code inputs is one, linear complex field codes yield the minimum union bound, for all SNR and for all input symbol alphabets.

5.4.2 An Optimum Encoding Matrix for $t = 4$, $N = K = 2$

Now, we consider the example of length $N = 2$ with $K = 2$ inputs operating over a $t = 4$-input channel, with an arbitrary number of outputs. Note that this combination satisfies the condition (69) of Proposition 3, and so, we can find $K = 2$ folded columns $\mathbf{M}_1$ and $\mathbf{M}_2$, which achieve the OD bound. To do so, we need to pick $NK = 4$ orthogonal vectors of dimension 4×1 (in general, $t \times 1$) to use as the columns of $\mathbf{M}_1$ and $\mathbf{M}_2$. For example, we can pick them as proportional to the columns of the identity matrix. Scaling these to ensure each 4×2 ($t \times N$) folded column has energy one, we get the following.

$$\mathbf{M}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{M}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{70}$$

The encoding matrix $\mathbf{M}$ of dimension 8×2 ($Nt \times K$) corresponding to these folded columns achieves the minimum possible union bound among all matrices in $\mathcal{M}(8, 2)$.

5.4.3 How Special are the Special Cases: A Rate-Diversity View

Proposition 3 solves the union bound optimization problem only when $K \leq \max(t/N, N/t)$. We will now see that this is a very restrictive condition. From the rank rule, the maximum raw diversity order of a length-N space-time code is $\delta = r\min(t, N)$. Two cases arise, depending on the relation between N and t.

If $N \geq t$, δ is equal to the full raw diversity order, namely tr. However, in this case, $N \geq t$ also implies that $\max(t/N, N/t) = N/t$. Now, Proposition 3 solves the optimization problem only when the number of space-time code inputs $K \leq N/t$, implying that the rate K/N can at most equal $1/t$. Clearly, this is a far cry from the full rate, namely $\min(t, r)$.

On the other hand, $N < t$ leads to a less-than-full diversity order of $\delta = Nr$. Now, Proposition 3 holds only when $K \leq t/N$. In other words, the rate K/N should be less than or equal to t/N^2. Substituting $N = \delta/r$, the maximum rate for which Proposition 3 applies is tr^2/δ^2. The following remark sums up the extent to which $K \leq \max(t/N, N/t)$ restricts the rate and raw diversity order, for which the optimization problem is solved.

> **Remark 4.** There is a trade-off between the rate and raw diversity of the optimum space-time codes that can be found by Proposition 3. From the rank rule, the raw diversity order δ always belongs to the set $\{r, 2r, \ldots, tr\}$. If one aims to achieve a raw diversity order of δ, then the maximum rate of the optimum space-time code that can be found using Proposition 3 equals
>
> $$R_{\max} = tr^2/\delta^2. \tag{71}$$

Recall from the last chapter that there is no fundamental tradeoff between rate and raw diversity order. In particular, linear space-time codes can simultaneously have any rate up to t and any raw diversity order upto tr. However, Proposition 3 finds the optimum encoding matrix only for space-time codes with the narrow range of rates and diversities given by Remark 4. In particular, it does not find the optimum matrix for the holy grail of code design, namely codes with both full rate and full raw diversity order.

The fundamental problem here is that strongly orthogonal folded columns cannot be found for $K > \max(t/N, N/t)$. We notice a loophole in Lemma 4, which says strong orthogonality of folded columns is only a *sufficient* condition for achieving the optimum OD bound. Therefore, one might hope to achieve the OD bound, even when the folded

columns are not strongly orthogonal. However, we will now show that, for a wide class of difference alphabets, strong orthogonality is in fact a *necessary* condition for reaching the OD bound, implying that the OD bound is unreachable when $K > \max(t/N, N/t)$.

5.5 The OD Bound Is Unreachable for Many Code Parameters

Recall that the input difference alphabet $\mathcal{D}$ is a set consisting of $K \times 1$ complex vectors. In this section, we focus on alphabets $\mathcal{D}$ with the following separability properties:

(i) For every integer $i \in \{1, 2, \ldots, K\}$, there is a difference vector $\mathbf{d} \in \mathcal{D}$ such that only its i^{th} element d_i is non-zero, but all other elements are zero.

(ii) For every pair of distinct integers $(i, j) \in \{1, 2, \ldots, K\}$, there are some two vectors $\mathbf{d}'$ and $\mathbf{d}''$ in $\mathcal{D}$ such that their i^{th} and j^{th} elements are non-zero, but all other elements are zero. Further, the non-zero elements satisfy $\text{Imag}(d_i' d_j'^{*} d_i'' d_j''^{*}) \neq 0$.

Properties (i) and (ii) essentially mean that there are difference vectors of Hamming weight one and two respectively, and the positions of the non-zero elements can be arbitrarily chosen. The condition $\text{Imag}(d_i' d_j'^{*} d_i'' d_j''^{*}) \neq 0$ in property (ii) is one way of ensuring that $\mathbf{d}'$ and $\mathbf{d}''$ are not proportional, hence $\mathbf{d}''$ does place some new constraint (that was not already placed by $\mathbf{d}'$) on the choice of the encoding matrix.

A wide class of difference alphabets have both the above properties. For example, consider the case where each of the K inputs to a space-time code is drawn independently from a complex QAM or PSK constellation. The difference alphabet $\mathcal{D}$ is just a cross-product of K copies of the individual symbol difference alphabet, consisting of the set of

all pairwise difference between valid QAM or PSK signals, respectively. It is easy to check that this alphabet satisfies both (i) and (ii). Having defined the alphabet properties, we can proceed to prove the following necessary condition.

Lemma 5. Suppose the input difference alphabet $\mathcal{D}$ to a strictly linear space-time code has properties (i) and (ii) above. Then, the orthogonal differences bound (63) is achieved only if the K folded columns $\{\mathbf{M}_i\}$ of the encoding matrix are strongly orthogonal, i.e., they satisfy (66), repeated below for convenience

$$\mathbf{M}_i^* \mathbf{M}_j = \frac{1}{N} \delta_{ij} \mathbf{I}_N \text{ if } t \geq N, \qquad \mathbf{M}_i \mathbf{M}_j^* = \frac{1}{t} \delta_{ij} \mathbf{I}_t \text{ if } t < N. \tag{72}$$

Proof: For simplicity, we will prove for the case $t \geq N$. In this case, (64) states that the OD bound is achieved if and only if $\mathrm{mat}(\mathbf{Md})^*\mathrm{mat}(\mathbf{Md}) = (\|\mathbf{d}\|^2/N)\, \mathbf{I}_N$ for all $\mathbf{d} \in \mathcal{D}$. We will show that this is possible only if $\{\mathbf{M}_i\}$ satisfies (72).

First, since $\mathcal{D}$ satisfies property (i), there is a difference vector $\mathbf{d}$ such that only the symbol d_i is non-zero. In this case, $\mathrm{mat}(\mathbf{Md}) = \mathbf{M}_i d_i$, and so $\mathrm{mat}(\mathbf{Md})^*\mathrm{mat}(\mathbf{Md}) = |d_i|^2 \mathbf{M}_i^* \mathbf{M}_i$. On the other hand, (64) requires $\mathrm{mat}(\mathbf{Md})^*\mathrm{mat}(\mathbf{Md}) = (\|\mathbf{d}\|^2/N)\, \mathbf{I}_N = (|d_i|^2/N)\, \mathbf{I}_N$. Equating the two expressions, we see that $\mathbf{M}_i$ should satisfy

$$\mathbf{M}_i^* \mathbf{M}_i = (1/N)\, \mathbf{I}_N, \text{ for all } i. \tag{73}$$

Next, we will use the fact that $\mathcal{D}$ also satisfies property (ii). For any pair (i, j), consider the difference vector $\mathbf{d}'$ where only the two elements d_i' and d_j' are non-zero. Clearly, $\mathrm{mat}(\mathbf{Md}') = \mathbf{M}_i d_i' + \mathbf{M}_j d_j'$. From this, we get

$$\mathrm{mat}(\mathbf{Md}')^*\mathrm{mat}(\mathbf{Md}') = (\mathbf{M}_i d_i' + \mathbf{M}_j d_j')^* (\mathbf{M}_i d_i' + \mathbf{M}_j d_j') \tag{74}$$

$$= |d_i'|^2 \mathbf{M}_i^* \mathbf{M}_i + |d_j'|^2 \mathbf{M}_j^* \mathbf{M}_j + d_i'^* d_j' \mathbf{M}_i^* \mathbf{M}_j + d_i' d_j'^* \mathbf{M}_j^* \mathbf{M}_i. \tag{75}$$

We anyway need $\mathbf{M}_i^* \mathbf{M}_i = \mathbf{M}_j^* \mathbf{M}_j = (1/N)\, \mathbf{I}_N$ from (73). Using this, (75) becomes

$$\mathrm{mat}(\mathbf{Md}')^* \mathrm{mat}(\mathbf{Md}') = \mathbf{I}_N\, (|d_1'|^2 + |d_2'|^2)/N + d_i'^* d_j' \mathbf{M}_i^* \mathbf{M}_j + d_i' d_j'^* \mathbf{M}_j^* \mathbf{M}_i. \qquad (76)$$

Note that $(|d_1'|^2 + |d_2'|^2) = \|\mathbf{d}'\|^2$. Thus, if $\mathrm{mat}(\mathbf{Md}')^* \mathrm{mat}(\mathbf{Md}')$ in (76) is to satisfy (64), it is necessary that $\mathbf{M}_i$ and $\mathbf{M}_j$ satisfy

$$d_i'^* d_j' \mathbf{M}_i^* \mathbf{M}_j + d_i' d_j'^* \mathbf{M}_j^* \mathbf{M}_i = \mathbf{0}. \qquad (77)$$

Applying the above argument to the other difference vector $\mathbf{d}''$, which is guaranteed to exist by property (ii), it is also necessary that

$$d_i''^* d_j'' \mathbf{M}_i^* \mathbf{M}_j + d_i'' d_j''^* \mathbf{M}_j^* \mathbf{M} = \mathbf{0}. \qquad (78)$$

Solving (77) and (78) as simultaneous equations, we get $\mathrm{Imag}(d_i' d_j'^* d_i'' d_j''^*) \mathbf{M}_i^* \mathbf{M}_j = \mathbf{0}$. Since property (ii) assures us that $\mathrm{Imag}(d_i d_j^* d_i' d_j'^*) \neq 0$, this implies

$$\mathbf{M}_i^* \mathbf{M}_j = \mathbf{0} \qquad (79)$$

for all distinct pairs (i, j). To sum up, we see that the two necessary conditions for the OD bound to be reached are (73) and (79). Clearly, these two amount to $\mathbf{M}_i^* \mathbf{M}_j = \frac{1}{N} \delta_{ij} \mathbf{I}_N$, as claimed in (72). This proves the lemma.

The above lemma states that under the stated conditions, the OD bound is achieved only if the K folded columns $\{\mathbf{M}_i\}$ are strongly orthogonal. On the other hand, Proposition 3 of the last section says that K strongly orthogonal folded columns cannot be found if $K > \max(t/N, N/t)$. Combining these two, we get the following result.

Remark 5. Consider a strictly linear space-time code of length N with K inputs operating over a t-input, r-output Rayleigh fading channel. If $K > \max(t/N, N/t)$ and the input difference alphabet $\mathcal{D}$ has the separation properties (i) and (ii), then it is

impossible to find an encoding matrix with orthonormal columns that achieves the orthogonal differences bound (53).

The result stated in Remark 5 is not surprising. Recall that in order to achieve the OD bound, the folded columns $\{\mathbf{M}_i\}$ should be chosen so that $\mathrm{mat}(\mathbf{Md})$ has equal singular values for all difference vectors $\mathbf{d} \in \mathcal{D}$. Thus, each difference vector $\mathbf{d} \in \mathcal{D}$ imposes a (possibly new) constraint, that should be accounted for while choosing $\{\mathbf{M}_i\}$. Suppose t and N are fixed. As K increases, the number of constraints on $\{\mathbf{M}_i\}$ increases in two ways. Firstly, a greater number of folded columns, namely K, needs to be chosen from the fixed set of all $t \times N$ matrices. Secondly, the number of difference vectors in $\mathcal{D}$, and hence the number of constraints to be satisfied by the chosen matrices $\{\mathbf{M}_i\}$ increases. The previous section suggested that as long as $K \leq \max(t/N, N/t)$, the constraints imposed by the difference vectors still leaves some freedom in the choice of $\{\mathbf{M}_i\}$. On the other hand, Remark 5 says that if $K > \max(t/N, N/t)$, the constraints overwhelm the available freedom, and the orthogonal differences bound cannot be satisfied.

5.6 Summary and Conclusions

We began this chapter by deriving an expression (52) for the union bound on word error rate, which was a sum over the difference alphabet $\mathcal{D}$. The orthogonal differences bound was obtained by bounding each term in the union bound by its lowest possible value, which is reached when $\mathrm{mat}(\mathbf{Md})$ has equal singular values. For the special case $K \leq \max(t/N, N/t)$, we constructively obtained encoding matrices which achieve the OD bound, thus solving the optimization problem. However, the requirement $K \leq \max(t/N, N/t)$ was shown to limit either the rate or the raw diversity order of the space-time code.

Since both high rate and raw diversity order are desirable, we would like to find space-time codes that minimize the union bound for the case $K > \max(t/N, N/t)$. However, for this case, we showed that the OD bound is unreachable for a wide class of input alphabets. This class includes the common input alphabet, consisting of independently modulated QAM input symbols. The fact that the OD bound is unreachable implies that all the individual terms in the union bound summation cannot *simultaneously* be minimized. Instead, optimum encoding matrices (infinite of which exist according to Section 5.2) optimally trade off the various terms in order to minimize the sum. In order to find these optimum encoding matrices, we will develop approximate numerical optimization techniques in the next chapter. We close this chapter by stating an interesting open problem. This can be skipped without affecting the readability of subsequent chapters.

5.6.1 Optimum Modulation for Space-Time Codes: An Open Problem

As discussed in Section 2.4.3, modulation (mapping bits to complex input symbols) and space-time encoding should ideally be designed jointly. However, joint design is analytically difficult, and has not yielded fruitful results in the literature, to the best of our knowledge. Instead, the standard simplifying assumption is that modulation and space-time code are done independently. Further, the modulation alphabet is typically some standard alphabet like QAM or PSK. In this section, we will argue that even if the modulation is done independently, one can minimize the achievable union bound by carefully selecting the modulation alphabet.

First, take the case of codes with $K \leq \max(t/N, N/t)$. We have seen that the OD bound

$$\sum_{\mathbf{d} \in \mathcal{D}} p_{\mathcal{D}}(\mathbf{d}) \left(1 + \frac{S\|\mathbf{d}\|^2}{4E_{\text{tx}}\min(t, N)}\right)^{-r\min(t, N)}, \tag{80}$$

can be achieved in this case. Crucially, by carefully choosing the input alphabet $\mathcal{D}$, one can minimize the OD bound (80). More precisely, given the size and the energy of the input alphabet (thus fixing the data rate and transmit energy respectively), the modulation problem is to find the input alphabet $\mathcal{U}$ whose corresponding difference alphabet minimizes the right hand side of (80) above. Note that (80) depends only on the Euclidean distance $\|\mathbf{d}\|$ between input vectors. Thus, the problem stated above is similar to the lattice coding [45][46] problem, hence we expect that lattice coding ideas can be used here.

In the case $K > \max(t/N, N/t)$, the modulation problem becomes more complicated, because the OD bound may or not may not be reachable, depending on the choice of the input alphabet. Note that Remark 5 proves that the orthogonal differences bound is unreachable only for difference alphabets satisfying the separation properties (i) and (ii). If these properties are not satisfied, the OD bound could potentially be reached. Thus, the modulation problem for the case $K > \max(t/N, N/t)$ would have to address both the reachability of the OD bound, and the choice of the input alphabet to reduce the OD bound itself. An alternative would be to not use the OD bound, but derive an expression for the lowest achievable union bound and minimize it directly.

We believe the optimization of the input alphabet could be a productive area of future research. However, for the rest of this work, we will assume that some choice of the input alphabet has been made, and aim to optimize the encoding matrix alone.

CHAPTER 6

Numerical Optimization of the Union Bound and Coding Gain

The answer to the Great Question of Life, the Universe, and Everything is. . . Forty-two.
— Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Try to prove by induction, construction, contradiction, or obfuscation. If all else fails, prove by MATLAB.
— *Old Jungle Saying*

In the previous chapter, we saw the analytical difficulty of obtaining encoding matrices for linear space-time codes that minimize the union bound. In this chapter, we will resort to an approximate numerical solution, obtained by viewing union bound optimization as a constrained optimization problem. We develop an optimization algorithm and use it to obtain space-time codes with near-optimum union bound, and consequently lower error rate than unoptimized codes. We also propose heuristic techniques to speed up the numerical algorithm. A truncated version of these results was presented in [42].

6.1 Code Design as a Constrained Optimization Problem

We begin this chapter by stating code design as a constrained optimization problem. The primary optimization metric or cost function is the union bound, but we will also consider coding gain. The constraint is to fix the transmit energy.

From (31) in Chapter 3, we see that the union bound on the word error rate of a strictly linear space-time code with input alphabet $\mathcal{U}$ operating over a t-input, r-output Rayleigh fading channel is

$$\sum_{\mathbf{u} \in \mathcal{U}} p_{\mathcal{U}}(\mathbf{u}) \sum_{\mathbf{u}' \neq \mathbf{u}} \det{}^{-r}\left(\mathbf{I}_t + S\frac{\mathrm{mat}(\mathbf{Md})(\mathrm{mat}(\mathbf{Md}))^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right). \tag{81}$$

In the remainder of this chapter, we will assume that each input symbol to the space-time code, i.e., each element of the $K \times 1$ input vector $\mathbf{u}$ is drawn *uniformly* from a finite, zero-mean alphabet with average energy one. Also, different symbols are drawn independently of each other. Under these assumptions, we now proceed to derive convenient representations of the transmit energy constraint, and computationally simple expressions for the union bound and the coding gain.

6.1.1 The Optimization Constraint

Note that the union bound (81) depends on the average transmit energy per signaling interval. Since the composite transmit vector $\mathbf{x} = \mathbf{Mu}$ contains N transmit vectors, the average transmit energy is

$$E_{\mathrm{tx}} = \frac{1}{N}\mathrm{E}[||\mathbf{x}||^2] = \frac{1}{N}\mathrm{E}[||\mathbf{Mu}||^2]. \tag{82}$$

We have assumed that each element of $\mathbf{u}$ is zero-mean, has unit energy and is independent of the other elements. It is easy to see that this implies $\mathrm{E}[||\mathbf{Mu}||^2] = ||\mathbf{M}||_{\mathcal{F}}^2$, giving $E_{\mathrm{tx}} = ||\mathbf{M}||_{\mathcal{F}}^2/N$. Thus, the average transmit energy is proportional to the energy $||\mathbf{M}||_{\mathcal{F}}^2$ of the $Nt \times K$ encoding matrix $\mathbf{M}$. One way to ensure fair comparison across different encoding matrices is to just substitute $E_{\mathrm{tx}} = ||\mathbf{M}||_{\mathcal{F}}^2/N$ into the union bound expression, and perform unconstrained optimization to seek matrices minimizing the union bound. Instead, we

choose to explicitly constrain the energy of $\mathbf{M}$ to be equal to the number of columns, namely K. Thus, we now seek an encoding matrix satisfying the energy constraint

$$\|\mathbf{M}\|_{\mathcal{F}}^2 = \mathrm{trace}(\mathbf{M}^*\mathbf{M}) = K \tag{83}$$

that minimize the union bound. Note that this fixes the average transmit energy to be $E_{\mathrm{tx}} = K/N$, which is also the rate of the space-time code. The advantage of an explicit constraint is that only one term in the cost function now depends on the encoding matrix. Also, this form enables easy comparison with existing space-time codes like the LCF codes, all of which assume normalized energy for $\mathbf{M}$.

Another way to view the energy constraint is that we are now only interested in finding optimum encoding matrix in the *constraint set*

$$\mathcal{L}(Nt, K) = \{Nt \times K \text{ matrices } \mathbf{M}\text{: } \mathrm{trace}(\mathbf{M}^*\mathbf{M}) = K\}. \tag{84}$$

Placing further constraints on the encoding matrix yields smaller constraint sets. For example, consider the set $\mathcal{M}(Nt, K)$ analyzed in the previous two chapters, containing all $Nt \times K$ matrices with orthonormal columns. Note that all matrices in $\mathcal{M}(Nt, K)$ have energy K, but not all energy K matrices have orthonormal columns. Thus,

$$\mathcal{M}(Nt, K) \subseteq \mathcal{L}(Nt, K) \tag{85}$$

holds, with equality if and only if $K = 1$.

Reducing the size of the constraint set makes the search space smaller, and the optimization search faster. On the other hand, it removes some potentially optimum matrices from consideration. In the case of $\mathcal{M}(Nt, K)$, we will show by simulation that the acceleration of the search does not come at the cost of optimality. In fact, all the optimum matrices we obtained by numerical methods will turn out to be in the set $\mathcal{M}(Nt, K)$.

6.1.2 Optimization Metrics: Union Bound and Coding Gain

Having discussed the optimization constraint, we now proceed to obtain easy-to-compute expressions for the optimization metrics, namely the union bound and coding gain. In the union bound expression (81), we can now substitute the following consequences of our assumptions so far. Firstly, because each symbol alphabet is finite, the input alphabet has a finite size $|\mathcal{U}| < \infty$. Secondly, since each symbol is drawn uniformly, $p_{\mathcal{U}}(\mathbf{u})$ is always $1/|\mathcal{U}|$ for all the valid input vectors to the space-time code. Thirdly, because of the energy constraint $E_{\text{tx}} = K/N$. Using these, (81) becomes

$$P_{\text{UB}}(S, \mathbf{M}) = \frac{1}{|\mathcal{U}|} \sum_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{u}' \neq \mathbf{u}} \det{}^{-r}\left(\mathbf{I}_t + \frac{NS}{4K}(\text{mat}(\mathbf{Md})(\text{mat}(\mathbf{Md}))^{\text{H}})\right), \qquad (86)$$

where $\mathbf{d} = \mathbf{u} - \mathbf{u}'$.

The computation of (86) involves a double sum over the input alphabet, but can be made more efficient using two simple observations. Firstly, note that the summand depends only on the difference vector $\mathbf{d} = \mathbf{u} - \mathbf{u}'$. Thus, all pairs $(\mathbf{u}, \mathbf{u}')$ with the same difference vector make the same contribution to the union bound, and can be treated identically. (This was done in Chapter 5 too, except we did not explicitly use the finiteness of the input alphabet.) Secondly, if two difference vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ are proportional by a unit-magnitude complex number, i.e., if $\mathbf{d}_2 = e^{j\phi}\mathbf{d}_1$, then it is easy to see that $\text{mat}(\mathbf{Md}_2)(\text{mat}(\mathbf{Md}_2))^* = \text{mat}(\mathbf{Md}_1)(\text{mat}(\mathbf{Md}_1))^*$. Consequently, $\mathbf{d}_1$ and $\mathbf{d}_2$ make the same contribution to (86). In other words, two difference vectors which are proportional and have the same magnitude, are equivalent with respect to the union bound, and can be said to belong to the same *equivalence class*. Combining these two observations, one evaluates the union bound over a compressed difference alphabet $\mathcal{C}$, consisting of one

representative vector from each equivalence class of difference vectors. Associated with each $\mathbf{d} \in \mathcal{C}$ is a multiplicity term $n_{\mathbf{d}}$, containing the number of pairs ($\mathbf{u}$, $\mathbf{u}'$) whose difference is equivalent to $\mathbf{d}$, i.e., $\mathbf{u} - \mathbf{u}' = e^{j\phi}\mathbf{d}$. Then, the union bound (86) becomes

$$P_{\mathrm{UB}}(S, \mathbf{M}) = \frac{1}{|\mathcal{U}|} \sum_{\mathbf{d} \in \mathcal{C}} n_{\mathbf{d}} \det{}^{-r}\left(\mathbf{I}_t + \frac{NS}{4K} \frac{\mathrm{mat}(\mathbf{Md})(\mathrm{mat}(\mathbf{Md}))^{\mathrm{H}}}{4E_{\mathrm{tx}}}\right). \tag{87}$$

Equivalently, writing each determinant in terms of the singular values of mat($\mathbf{Md}$) gives

$$P_{\mathrm{UB}}(S, \mathbf{M}) = \frac{1}{|\mathcal{U}|} \sum_{\mathbf{d} \in \mathcal{C}} n_{\mathbf{d}} \prod_{i=1}^{\mathrm{rank}(\mathrm{mat}(\mathbf{Md}))} \left(1 + \frac{NS}{4K}\lambda_i^2\right)^{-r}. \tag{88}$$

In addition to the union bound, we will also consider the coding gain as an optimization metric. In the above notation, it is easy to see that the coding gain (39) is given by

$$\gamma_{\mathrm{CG}}(\mathbf{M}) = \min_{\substack{\mathbf{d} \in \mathcal{C} \\ \mathrm{rank}(\mathrm{mat}(\mathbf{Md})) = \rho_{\min}}} \prod_{i=1}^{\mathrm{rank}(\mathrm{mat}(\mathbf{Md}))} \left(\frac{N}{4K}\lambda_i^2\right)^{r}. \tag{89}$$

Both the union bound and the coding gain depend on the encoding matrix $\mathbf{M}$. However, only the union bound depends on the SNR S. The coding gain implicitly assumes asymptotically high SNR.

6.1.3 Constrained Optimization Problem Statement

We can now precisely state the optimization problem.

Code Design Problem Statement: Given the number of transmit antennas t, the number of receive antennas r, the code length N, the number of inputs K, the finite input alphabet $\mathcal{U}$, and the SNR S, find an $Nt \times K$ complex encoding matrix $\mathbf{M}$

belonging to the constraint set $\mathcal{S}(Nt, K)$ that has the optimum value of the chosen optimization metric.

The optimization metric is either the union bound (87)(88) or the coding gain (89). The former needs to be minimized, while the latter needs to be maximized. The constraint set $\mathcal{S}(Nt, K)$ is some well-chosen subset of the energy constrained set $\mathcal{L}(Nt, K)$ (84). The rest of this chapter is devoted to solving the above problem.

6.2 Numerical Solution to the General Optimization Problem

In this section, we propose an approximate numerical solution to the optimization problem. The numerical solution proceeds in two stages, namely random search and gradient descent search. In the first stage, a large number, say N_{rand}, of encoding matrices is generated from the constraint set. The matrix with the best optimization metric is picked as the initial matrix $\mathbf{M}^{(0)}$ for the second stage, namely the iterative gradient descent stage. In iteration k, the matrix $\mathbf{M}^{(k)}$ is computed by making a slight, well chosen, shift to the previous matrix $\mathbf{M}^{(k-1)}$, and *rounding off* the resultant matrix back to the closest matrix in the constraint set. In other words, each iteration of gradient descent is

$$\tilde{\mathbf{M}}^{(k-1)} = \mathbf{M}^{(k-1)} + \Delta^{(k-1)}, \tag{90}$$

$$\mathbf{M}^{(k)} = \underset{\mathbf{M} \in \mathcal{S}(Nt, K)}{\operatorname{argmin}} \; \|\tilde{\mathbf{M}}^{(k-1)} - \mathbf{M}\|_{\mathcal{F}}, \tag{91}$$

where (90) performs the shift operation and (91) rounds off the result.

The iterative process is repeated till either a pre-determined number of iterations N_{gd} have been performed, or the process converges, i.e., $\|\mathbf{M}^{(k)} - \mathbf{M}^{(k-1)}\|_{\mathcal{F}}$ falls below a threshold ε. Typical values are $N_{\text{rand}} = 10000$, $N_{\text{gd}} = 2500$ and $\varepsilon = 10^{-6}$. We now fill in precise details about the gradient descent step, particularly the choice of the shift $\Delta^{(k-1)}$, and the nature of the rounding-off operation in each iteration.

6.2.1 The Shift Operation in Iterative Gradient Descent

There are some slight differences in the shift matrices for union bound and coding gain optimization. We first take up the union bound, which from (87), is a smooth, continuous analytic function of the elements of $\mathbf{M}$. It can be differentiated with respected to each element of $\mathbf{M}$, yielding functions

$$g_{ij}(S, \mathbf{M}) = \frac{\partial}{\partial m_{ij}} P_{\text{UB}}(S, \mathbf{M}). \tag{92}$$

Expressions for $g_{ij}(S, \mathbf{M})$ are computed in Appendix B. In iteration k of gradient descent, the shift matrix $\Delta^{(k-1)}$ is assembled, according to

$$\Delta^{(k-1)}_{ij} = -\mu g_{ij}(S, \mathbf{M}^{(k-1)}). \tag{93}$$

In other words, each element of $\Delta^{(k-1)}$ is proportional of the derivative of $P_{\text{UB}}(S, \mathbf{M})$ with respect to the corresponding element of $\mathbf{M}$, evaluated at $\mathbf{M} = \mathbf{M}^{(k-1)}$. The negative sign in the constant of proportionality, $-\mu$, forces a move *opposite* to the gradient, since we are interested in decreasing the value of the union bound. The constant μ itself is chosen heuristically. It is typically around 0.1.

Now, we turn our attention to the other possible optimization metric, namely the coding gain. Here, computing the shift matrix is complicated by the fact that the coding gain (89) is not always an analytic function of the elements of $\mathbf{M}$, i.e., at some values of $\mathbf{M}$, its gradient is different along different directions. The reason is that $\gamma_{\mathrm{CG}}(\mathbf{M})$ is the point-wise minimum of many smooth, continuous functions, namely

$$\gamma_{\mathbf{d}}(\mathbf{M}) = \prod_{i=1}^{\mathrm{rank}(\mathbf{M}\mathbf{d})} \left(\frac{N\lambda_i^2}{4K}\right)^r \quad , \tag{94}$$

for each difference vector $\mathbf{d} \in \mathcal{C}$.

In any iteration k, consider the calculation of the coding gain $\gamma_{\mathrm{CG}}(\mathbf{M}^{(k-1)})$ by taking the minima of $\gamma_{\mathbf{d}}(\mathbf{M}^{(k-1)})$. Two cases arise. Either there is exactly one function $\gamma_0(\mathbf{M}^{(k-1)})$ which is less than all the other functions, or there are multiple functions with the same minimum value. In the former case, the coding gain $\gamma_{\mathrm{CG}}(\mathbf{M})$ is equal to the analytic function $\gamma_0(\mathbf{M})$ in a neighborhood around $\mathbf{M}^{(k-1)}$ and is therefore itself analytic at $\mathbf{M}^{(k-1)}$. In this case, one can define the shift matrix for gradient descent by

$$\Delta^{(k-1)}_{ij} = \mu \frac{\partial}{\partial m_{ij}} \gamma_0(\mathbf{M}^{(k-1)}). \tag{95}$$

Note that, in contrast to the shift matrix for the union bound (93), there is no negative sign here. This is because the coding gain needs to be maximized, while the union bound needs to be minimized. The shift attempts to maximize the coding gain by *moving along* the gradient, which is by definition, the direction in which $\gamma_{\mathrm{CG}}(\mathbf{M})$ increases most rapidly.

The problem with the coding gain arises in the second case, where there are multiple coincident minima at the point $\mathbf{M}^{(k-1)}$, say $\gamma_1(\mathbf{M}^{(k-1)}) = \gamma_2(\mathbf{M}^{(k-1)}) = \ldots \gamma_L(\mathbf{M}^{(k-1)})$. Each of these functions has its own gradient, or equivalently, a distinct direction of maximum

increase. Consequently, it is not clear which is the direction of maximum increase for the minimum of these functions, $\gamma_{CG}(\mathbf{M})$. One way to do this is to move along a direction which has equal projections on the gradients of all the $\{\gamma_l(\mathbf{M}^{(k-1)})\}$. This is possible if and only if the individual gradients are linearly independent. On the other hand, when the gradients of the individual functions are linearly dependent, the only thing we could think of was to stop the gradient descent process.

6.2.2 The Rounding Off Operation in Iterative Gradient Descent

Recall that the rounding off operation needs to find the matrix $\mathbf{M}^{(k)}$, which is at the minimum Euclidean distance to the shifted matrix $\tilde{\mathbf{M}}^{(k-1)}$, among all matrices in the constraint set $\mathcal{S}(Nt, K)$. This computation, would of course, depend on the constraint set itself. We will state, without proof, the computation formula, for each constraint set.

For the energy-constrained set $\mathcal{L}(Nt, K)$, the closest matrix is obtained by just scaling $\tilde{\mathbf{M}}^{(k-1)}$ to have energy K. Thus,

$$\mathcal{S}(Nt, K) = \{\mathbf{M}: \|\mathbf{M}\|_{\mathcal{F}}^2) = K\} \quad \Rightarrow \quad \mathbf{M}^{(k)} = \frac{\sqrt{K}}{\left\|\tilde{\mathbf{M}}^{(k-1)}\right\|_{\mathcal{F}}} \tilde{\mathbf{M}}^{(k-1)}. \tag{96}$$

On the other hand, the constraint set $\mathcal{M}(Nt, K)$ requires the columns to be orthonormal, or equivalently it requires the singular values of $\mathbf{M}^{(k)}$ to be equal. The rounding off here is done by just normalizing the singular values of $\tilde{\mathbf{M}}^{(k-1)}$, i.e.,

$$\mathcal{M}(Nt, K) = \{\mathbf{M}: \mathbf{M}^{\mathrm{H}}\mathbf{M} = \mathbf{I}_K\} \Rightarrow \quad \mathbf{M}^{(k)} = \frac{1}{\sqrt{K}} \tilde{\mathbf{U}}\tilde{\mathbf{V}}^*, \tag{97}$$

where $\tilde{\mathbf{M}}^{(k-1)} = \tilde{\mathbf{U}}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^*$ is the singular value decomposition of $\tilde{\mathbf{M}}^{(k-1)}$.

6.2.3 The Complexity of Numerical Optimization

One main concern about the above numerical algorithm is its high computational complexity. The main reason for this complexity is the intractable nature of the metrics themselves. For example, the union bound, in principle, needs a sum over all pairwise input difference vectors, whose number increases exponentially with the data rate. More precisely, since $\log_2|\mathcal{U}|$ information bits are transmitted N signaling intervals, the data rate of the space-time code, in b/s/Hz is given by $R_b = \log_2|\mathcal{U}|/N$. Given the required data rate R_b, the input alphabet size necessary to achieve the data rate, for a space-time code of length N, is $|\mathcal{U}| = 2^{NR_b}$. Ideally, one would like both high data rate, and a large length N, so as to achieve high raw diversity order $r\min(t, N)$ of the space-time code. Both factors lead to an exponential increase in the input alphabet size $|\mathcal{U}|$. Now, the size of the difference alphabet $\mathcal{D}$ is roughly the square of the input alphabet size. Using the structure in the union bound, and the proportionality of certain inputs, we obtained the compressed difference alphabet $\mathcal{C}$. This reduces the number of terms in the sum for the union bound (88). However, even the compressed alphabet $\mathcal{C}$ has size exponential in the data rate and code length, and hence, the computation of the union bound or the coding gain is a computationally daunting task.

The numerical algorithms compound the computational problem by requiring repeated summations over the alphabet $\mathcal{C}$. In the random search stage, the union bound needs to be computed for N_{rand} random matrices. In the gradient descent stage, the gradient matrix needs to be computed, again requiring a sum over the alphabet $\mathcal{C}$. Note that the coding gain is only marginally easier to compute than the union bound. Each vector $\mathbf{d} \in \mathcal{C}$ still needs to be processed, though the processing is simpler than the union bound, as seen by

comparing (89) to (88). The main advantage of the coding gain is that it is SNR-independent, while the union bound is SNR-dependent. Therefore, the coding gain needs to be optimized just once, whereas the union bound has to be re-optimized for every SNR. The price paid for the simplicity is that the coding gain reflects performance less faithfully than the union bound, since it focuses only on the worst case error event (see Chapter 3).

The main argument in favor of using the optimization algorithm is that it needs to be implemented just once. Once an optimum encoding matrix has been found for a given SNR, transmitter design is complete. The optimized encoding matrices so obtained significantly outperform un-optimized matrices, as confirmed by simulation results in the next section.

6.3 A Case Study of Union Bound Optimization: $t = r = N = 2, K = 4$

In this section, we run the numerical optimization algorithm for a specific optimization problem. We seek to optimize the union bound of a strictly linear space-time code encoding $K = 4$ complex inputs in blocks of length $N = 2$ signaling intervals over a Rayleigh fading channel with $t = 2$ inputs and $r = 2$ outputs. Each of the four inputs is drawn independently from a Gray-coded 4-QAM constellation, normalized to ensure unit energy. This is the same setup as the example discussed in Section 4.5 to illustrate that a randomly generated code yields the same full raw diversity order as an LCF code, but suffers a higher error rate than the LCF code. We now seek to further reduce the error rate by optimizing the union bound, with the SNR fixed at 23 dB. Note that the space-time codes under consideration have rate $K/N = 2$, which is equal to the full rate $\min(t, r)$ of the channel. Thus, the problem can be viewed as one of further optimizing linear space-

time codes which already have full rate and full raw diversity order. The results of numerical optimization for this specific example are also used to support some conjectures about optimum matrices for the general optimization problem.

6.3.1 Optimum Matrices Have Orthonormal Columns

First, the numerical optimization algorithm was used to search for optimum encoding matrices in the general energy constrained set $\mathcal{L}(4, 4)$, consisting of all 4×4 matrices with energy $K = 4$. Just to ensure convergence, the number of random matrices generated and the number of gradient descent iterations were both large, namely $N_{\text{rand}} = 200000$ and $N_{\text{gd}} = 60000$ respectively. The matrix $\mathbf{M}_{\mathcal{L}}$ obtained at the end of the search had a union bound of 1.386×10^{-4} at SNR 23 dB. In particular, the singular values of $\mathbf{M}_{\mathcal{L}}$ were found to be 1.0166, 1.0030, 0.9974 and 0.9927. We notice that these singular values are almost identical, indicating that $\mathbf{M}_{\mathcal{L}}$ is almost unitary, i.e., its columns are nearly orthonormal. This leads to the following conjecture.

> **Conjecture 1.** If the K inputs to a linear space-time code of length N operating over a t-input, r-output Rayleigh fading channel are drawn independently from a QAM constellation, the union bound is minimized by choosing an $Nt \times K$ encoding matrix with orthonormal columns.

An approximate justification of the conjecture is obtained by considering the orthogonal differences bound derived in Chapter 5, namely

$$P_{\text{UB}}(S, \mathbf{M}) \geq \sum_{\mathbf{d} \in \mathcal{D}} p_{\mathcal{D}}(\mathbf{d}) \left(1 + \frac{NS\|\mathbf{M}\mathbf{d}\|^2}{4K\min(t, N)}\right)^{-r\min(t, N)} . \tag{98}$$

We now argue that the OD bound is optimized by an encoding matrix $\mathbf{M}$ with orthonormal columns. Let the singular value decomposition of the encoding matrix be $\mathbf{M} = \mathbf{USV}^*$. Since $\mathbf{U}$ has orthonormal columns, $\|\mathbf{Md}\|^2 = \|\mathbf{USV}^*\mathbf{d}\|^2 = \|\mathbf{SV}^*\mathbf{d}\|^2$ for all $\mathbf{d}$, hence $\mathbf{U}$ can be chosen arbitrarily without affecting any term in the orthogonal differences bound. Choosing the $K \times K$ unitary matrix $\mathbf{V}$ *rotates* the difference alphabet $\mathcal{D}$. More precisely, defining the rotated alphabet $\mathcal{D}' = \{\mathbf{V}^*\mathbf{d}, \mathbf{d} \in \mathcal{D}\}$, the OD bound becomes

$$\sum_{\mathbf{d}' \in \mathcal{D}'} p_{\mathcal{D}}(\mathbf{Vd}') \left(1 + \frac{NS\|\mathbf{Sd}'\|^2}{4K\min(t, N)}\right)^{-r\min(t, N)}. \tag{99}$$

Note that the diagonal elements of the $K \times K$ diagonal matrix $\mathbf{S}$ are the singular values of $\mathbf{M}$, whose squared sum should equal the energy of $\mathbf{M}$, namely K. The effect of these singular values on the OD bound (99) is that they scale the corresponding elements of the vectors in $\mathcal{D}'$. The core of our argument is that it is optimal to scale all the elements identically. This is because the original difference alphabet $\mathcal{D}$ is a cross-product of individual QAM alphabet difference symbols, and is therefore roughly symmetric in space, i.e., the difference vectors in $\mathcal{D}$ are distributed roughly uniformly in K-dimensional complex space. Consequently, the rotated alphabet $\mathcal{D}'$ is also symmetric. In particular, each of its elements behaves roughly identically. Therefore, it is intuitively appealing to treat all these elements identically. Hence, we conjecture that all the singular values in $\mathbf{S}$ should be equal, implying that $\mathbf{M}$ should have orthonormal columns.

Conjecture 1 motivates a repeat of the optimization algorithm, restricting the search to the smaller orthonormal-column constraint set $\mathcal{M}(4, 4)$. In the first phase, $N_{\text{rand}} = 10000$ random matrices were generated, and the best one was used to initialize gradient descent with $N_{\text{gd}} = 10000$ iterations. This resulted in a matrix $\mathbf{M}_{\mathcal{M}}$ with a union bound $1.384 \times$

10^{-4} at 23 dB SNR. Comparing the results of the two searches, we see that $\mathcal{M}(4, 4)$ yields about the same union bound as $\mathcal{L}(4, 4)$ (1.386×10^{-4}). The advantage is that the search proceeds much faster in the smaller constraint set $\mathcal{M}(4, 4)$, since only 20000 total matrices were generated as against 260000 for $\mathcal{L}(4, 4)$.

6.3.2 A New Constraint Set for Length Two, Rate *t* Codes

Continuing with the approach of accelerating the search by focussing on smaller constraint sets, we now present a new constraint set for codes of length $N = 2$ with $K = 2t$ inputs. The encoding matrix of such codes has dimension $Nt \times K \equiv 2t \times 2t$. The proposed constraint set $\mathcal{N}(2t, 2t)$ contains all $2t \times 2t$ matrices of the form

$$\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{I}_t & e^{i\pi/4}\mathbf{I}_t \\ \mathbf{Q} & -e^{i\pi/4}\mathbf{Q} \end{bmatrix}, \tag{100}$$

where $\mathbf{I}_t$ is the $t \times t$ identity matrix, and $\mathbf{Q}$ is some $t \times t$ unitary matrix, i.e., $\mathbf{Q}^*\mathbf{Q} = \mathbf{Q}\mathbf{Q}^* = \mathbf{I}_t$. It is easy to check that all matrices of the form (100) are unitary, i.e., their columns are orthonormal. Also, any matrix in $\mathcal{N}(2t, 2t)$ is completely specified by the $t \times t$ unitary matrix $\mathbf{Q}$. Consequently, $\mathcal{N}(2t, 2t)$ is a small subset than $\mathcal{M}(2t, 2t)$, which contains all $2t \times 2t$ unitary matrices.

When the encoding matrix $\mathbf{M}$ belongs to $\mathcal{N}(2t, 2t)$, the encoding process can be visualized as follows. The input vector to the code $\mathbf{u}$ has dimension $2t$ ($= K$) $\times 1$. Split it into two $t \times 1$ halves, according $\mathbf{u}^T = [\mathbf{u}_1{}^T \ \mathbf{u}_2{}^T]$. Then, using the structure (100) of the encoding matrix, the composite transmit vector $\mathbf{x} = \mathbf{M}\mathbf{u}$ is given by

$$\frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{u}_1 + e^{i\pi/4}\mathbf{u}_2 \\ \mathbf{Q}(\mathbf{u}_1 - e^{i\pi/4}\mathbf{u}_2) \end{bmatrix}. \tag{101}$$

Equivalently, the encoder first scales the second half of the input vector to obtain $\mathbf{u}_2' = e^{i\pi/4}\mathbf{u}_2$. In the first signaling interval, the transmit vector is $\mathbf{u}_1 + \mathbf{u}_2'$. In the second signaling interval, the difference $\mathbf{u}_1 - \mathbf{u}_2'$ is rotated by a unitary matrix $\mathbf{Q}$ to obtain the second transmit vector $\mathbf{Q}(\mathbf{u}_1 - \mathbf{u}_2')$. A detailed description of how the structure (100) was obtained can be found in Appendix C.

Note that the example under consideration does have $N = 2$. Therefore, the optimization algorithm was run[1] for the constraint set $\mathcal{N}(4, 4)$. After generating only $N_{\text{rand}} = 3000$ random matrices, and $N_{\text{gd}} = 1000$ gradient descent iterations, the encoding matrix

$$\mathbf{M}^*_{2,2,2,4} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & e^{i\pi/4} & 0 \\ 0 & 1 & 0 & e^{i\pi/4} \\ 0.4456 & -0.8952i & -0.4456e^{i\pi/4} & 0.8952e^{i3\pi/4} \\ 0.8952i & -0.4456 & -0.8952e^{i3\pi/4} & 0.4456e^{i\pi/4} \end{bmatrix} \tag{102}$$

was obtained. The corresponding union bound at 23 dB was equal to 1.380×10^{-4}, which is the least among the matrices examined in this example. Thus, among the constraint sets examined for this problem, $\mathcal{N}(4, 4)$ yielded the best result fastest. We conjecture that for any general t, optimum matrices for length 2, rate t codes can be found in $\mathcal{N}(2t, 2t)$.

6.3.3 Optimizing Union Bound Reduces Error Rate

We now present simulation results to confirm that union bound optimization serves its ultimate purpose, namely the reduction of the error rate of space-time codes. The simulation in Section 4.5 is repeated, but now, the optimized encoding matrix $\mathbf{M}^*_{2,2,2}$ is compared to the LCF and random codes. The frame error rate is plotted against SNR in

1. Generating a random matrix in $\mathcal{N}(2t, 2t)$ amounts to generating the random $t \times t$ unitary matrix $\mathbf{Q}$. Also gradient descent is performed on $\mathbf{Q}$. Thus, the gradient matrix is calculated with respect to the elements of $\mathbf{Q}$. Rounding off the shifted matrix to the nearest $t \times t$ unitary matrix follows (97).

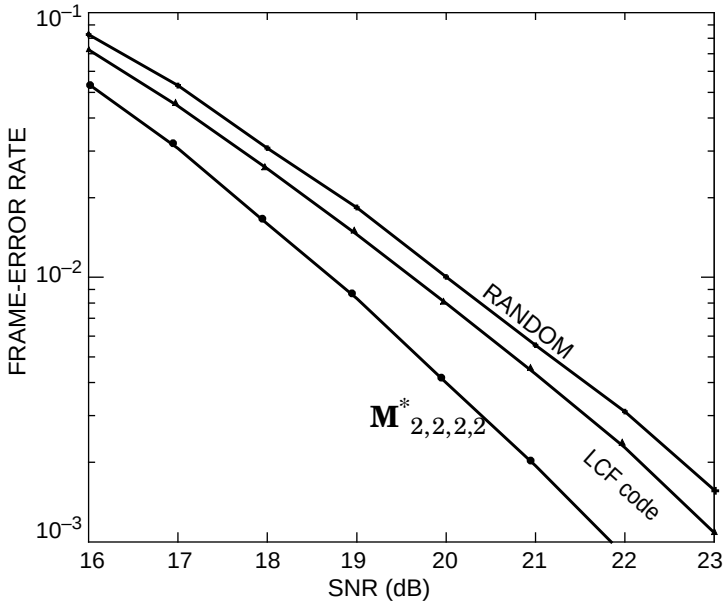

Fig. 6. Performance of three length-2, full-rate complex linear space-time codes over a 2-input, 2-output Rayleigh fading channel at 4 b / s / Hz.

Fig. 6. The LCF code was already seen to have a 0.5 dB performance advantage over the randomly generated code with orthonormal columns. The optimized code obtains a further advantage of 1.25 dB over the LCF code.

6.4 Another Case Study: $t = r = 2, N = 3, K = 6$

We now present one more case study of union bound optimization, again for 2-input, 2-output Rayleigh fading channel. Now, we consider codes with rate $R = 2$ (note that this is full rate) and length $N = 3$. Again, we assume that each of the $K = NR = 6$ code inputs is drawn from a 4-QAM input alphabet. Note that $N \neq 2$, so the constraint set $\mathcal{N}(2t, 2t)$ developed in the previous case cannot be used here.

We first performed a numerical search over all LCF codes, i.e., we tried different values of the unit-magnitude complex numbers α and β of the LCF encoding matrix in (20) of Section 2.5.3. For the choice $\alpha = \exp(i\pi/4)$ and $\beta = \exp(i\pi/32)$ suggested in [20], the union bound for this choice of parameters at 22.5 dB was computed to be $5.758 \times$

10^{-4}. Instead, after computing the union bound for 1000 random values of α and β, we found that the choice $\alpha = 0.575\pi$ and $\beta = 0.198\pi$ gives a union bound of 1.980×10^{-4} at 22.5 dB SNR. Then, instead of restricting the search to LCF code, we broadened the search to all encoding matrices belonging to the constraint set $\mathcal{M}(6, 6)$, namely all 6×6 unitary matrices. The numerical optimization procedure (random search followed by gradient descent) yielded the matrix

$$\mathbf{M}^*_{2,\,2,\,3,\,6} =$$

$$\frac{1}{\sqrt{3}} \begin{bmatrix} 1.1213e^{0.4800i} & 0.7645e^{0.7278i} & 0.7502e^{-1.0717i} & 0.3415e^{0.9279i} & 0.2240e^{0.9428i} & 0.6092e^{0.3002i} \\ 0.6935e^{-1.5064i} & 0.6893e^{-0.5880i} & 0.3364e^{-0.4479i} & 0.7437e^{1.3042i} & 0.6094e^{-1.2322i} & 0.9788e^{-0.9090i} \\ 0.5260e^{-0.3970i} & 0.6254e^{-0.8821i} & 0.7649e^{0.4154i} & 0.2369e^{-1.5205i} & 1.0556e^{-1.0312i} & 0.7837e^{0.9719i} \\ 0.6678e^{1.5201i} & 0.7445e^{-0.3741i} & 0.8046e^{-1.5060i} & 1.0235e^{0.1428i} & 0.4286e^{-1.2796i} & 0.3626e^{-1.2070i} \\ 0.3317e^{-0.9406i} & 0.6814e^{-0.8321i} & 0.7207e^{0.3717i} & 1.0348e^{0.2891i} & 0.8535e^{1.5501i} & 0.3412e^{0.7838i} \\ 0.6550e^{0.2066i} & 0.7285e^{1.0970i} & 0.7564e^{1.5465} & 0.3948e^{-0.0569i} & 0.7429e^{-1.2140i} & 0.8993e^{-0.4201i} \end{bmatrix} \quad (103)$$

with union bound 1.549×10^{-4} at 22.5 dB SNR.

It is clear that $\mathbf{M}^*_{2,\,2,\,3,\,6}$ has the least union bound among all matrices considered. We now present simulation results to confirm that this advantage also translates to a low word error rate. Frames consisting of consisting of 50 space-time code blocks, corresponding to $50N = 150$ signaling intervals over the channel were transmitted. As always, the channel is assumed to be constant over one frame, but varies independently from one from to the next. With ML decoding at the receiver, the resulting frame error rates of the three codes are plotted against SNR in Fig. 7. The optimized code $\mathbf{M}^*_{2,\,2,\,3,\,6}$ outperforms the unoptimized LCF code suggested in [20] by nearly 2 dB, and the optimized LCF code by 0.5 dB at a frame error rate of 3×10^{-2}.

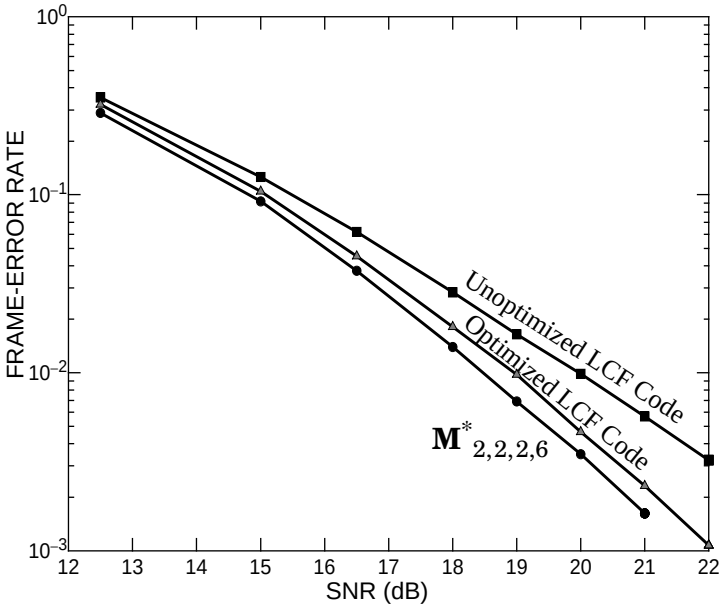

Fig. 7. Performance of three length-3, full-rate complex linear space-time codes over a 2-input, 2-output Rayleigh fading channel at 4 b / s / Hz.

Before concluding, we state the result of optimization for length 2, full rate code operating over a 3-input, 3-output Rayleigh fading channel, at an SNR of 19 dB with 4-QAM inputs. Since $N = 2$, the constraint set $\mathcal{N}(6, 6)$ can be used for fast search. For legibility, we present only the unitary portion $\mathbf{Q}$ in the general structure (100)

$$\mathbf{Q}_{3,3,2,6} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0.0228 + 0.1324i & -0.2762 + 0.9336i & -0.1842 - 0.0096i \\ 0.0560 + 0.0708i & 0.1545 + 0.0944i & 0.7342 - 0.6481i \\ -0.9168 + 0.3651i & 0.1364 + 0.0265i & 0.0800 - 0.0226i \end{bmatrix}. \qquad (104)$$

The above parameters correspond to a code with $K = 6$ inputs. As the number of inputs $K = NR$ increases, optimization becomes more computationally burdensome, primarily due to the difficulty in the computation of the union bound. Beyond $K = 10$ inputs, optimization is infeasible, even for the smallest complex input alphabet, namely 4-QAM.

6.5 Conclusions

In this chapter, we have discussed numerical optimization techniques to find encoding matrices that minimize the union bound, given the channel dimensions, encoding matrix dimensions and SNR. Simulation results show that such optimized codes yield a lower error rate than other unoptimized codes with the same rate and raw diversity order.

The drawback of the numerical optimization algorithm developed here is its high computational complexity for large code lengths and rates. One way to get around this problem would be to obtain a new optimization metric that is a faithful indicator of actual error rate, but is at the same time easier to compute than the union bound or the coding gain. Obtaining such a metric is an open problem. Another important open problem in the design of space-time codes is to analytically obtain a general structure for optimum encoding matrices.

This chapter ends our discussion of space-time codes with uncoded inputs. We now proceed to study the design of space-time inner codes, whose inputs are obtained from a powerful outer code. The new design problem calls for a new set of performance metrics, as we will in the next chapter.

CHAPTER 7

Introduction to Concatenated Space-Time Architectures

In Chapter 3, we derived performance metrics for stand-alone space-time codes, whose inputs were obtained by modulating uncoded bits. In subsequent chapters, we found linear space-time codes that optimize these performance metrics. In this chapter, we begin the analysis of the concatenated architecture shown in Fig. 8. A powerful outer code (such as turbo code [31] or LDPC [47][48]) produces coded bits, which are then modulated to form the input symbols to a space-time *inner* code. In this chapter, we will see that the optimized stand-alone space-time codes obtained in earlier chapters are not necessarily optimum space-time inner codes. We also describe the design methodology that will be used in subsequent chapters to find good space-time inner codes.

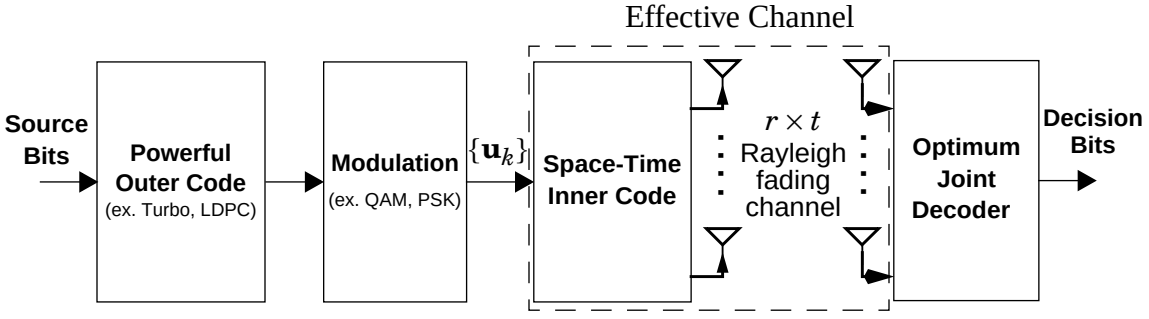

Fig. 8. Concatenated transmitter with optimum receiver.

7.1 The Need for New Performance Metrics

In this section, we will see that performance metrics derived for stand-alone space-time codes in Chapter 3 do not accurately reflect goodness of a space-time inner codes. To illustrate this fact, consider the case of the serial-to-parallel converter as a space-time code. As discussed in Section 2.5.1, the S/P converter, as a stand-alone space-time code, provides no transmit diversity while operating over a t-transmit, r-receive antenna Rayleigh fading channel. Intuitively, the reason for the lack of diversity is that each input symbol is transmitted from one transmit antenna alone. Consequently, if the signal from any antenna is wiped out due to deep channel fades, the symbols transmitted from that antenna cannot be recovered by the receiver, irrespective of how many other transmit antennas are present.

On the other hand, consider the same S/P converter as an inner code in Fig. 8. A well-designed outer code introduces redundancy across different signaling intervals and different antennas. Even if signals from one transmit antenna get wiped out, the joint decoder at the receiver can use the signals received from the other transmit antennas, and exploit the redundancy introduced by the outer code to estimate the lost signals. Thus, in the presence of a powerful outer code, each symbol does benefit from the presence of all t transmit antennas.

The example of the S/P converter illustrates the fact that the performance metrics of Chapter 3 do not accurately reflect actual error probabilities, in the presence of an outer code. This observation motivates the need for a different analytical approach to the concatenated architecture. We now present two possible approaches. The second one will be used in the rest of this work.

7.2 The Super-Code View of the Concatenated Architecture

The union bound and raw diversity order focus only on the space-time code, and fail to account for the outer code. A more comprehensive picture is obtained by treating the combination of the outer code and the space-time inner code as one large space-time supercode. The length $\overline{N}$ of this supercode is the number of signaling intervals required to transmit all the symbols in one codeword of the powerful outer code. For a powerful code, the length of the supercode satisfies $\overline{N} >> t$. From the rank rule (Section 3.3), one can achieve a diversity order of $r\min(t, \overline{N}) = tr$ by ensuring that all pairwise differences between the $t \times \overline{N}$ transmit matrices have full rank.

The codebook of the supercode (i.e., the set of all valid transmit matrices) can be directly analyzed to obtain exact or approximate expressions for the error probability of the complete system in Fig. 8. The analysis can be performed either for idealized optimum decoding [38], or for message-passing iterative decoding between soft-output decoders for the space-time inner and the outer code [49][50]. The analytically obtained error rate (or approximation thereof) can be used as a performance metric to evaluate and optimize both the outer code and the space-time inner code, or one given the other.

The problem with this approach is that an expression for the error probability of the supercode cannot always be found. Often, one needs to make some kind of idealized approximation about the outer code. For example, the analysis in [49][50] assumes that the outer code is a binary turbo or LDPC code of infinite length. The second design approach, which we discuss next, takes these approximations all the way, and assumes an ideal outer code.

7.3 Using the Effective Channel to Design Space-Time Inner Codes

Instead of grouping together the outer code and inner codes, suppose one considers the effective channel formed by the combination of the space-time code and MIMO fading channel. This effective channel is shown in dotted lines in Fig. 8. One signaling interval of the effective channel corresponds to one space-time code block, and lasts N signaling intervals across the MIMO fading channel. The union bound and raw diversity order of Chapter 3 measure the performance of the effective channel in one such block. In effect, they measure the error probability of uncoded transmission across the effective channel. However, in the concatenated architecture, the outer code codes across this effective channel, and makes its inputs dependent from one block to the next.

Recall from Section 2.4.1 that the choice of the space-time code completely determines the effective channel. In order to design a good space-time inner code, we need to quantify how well the effective channel responds to outer coding. Accounting for the actual outer code, and obtaining an error probability expression is difficult. Instead, one can replace the actual outer code by a hypothetical *capacity-achieving* outer code. Before proceeding, we point out that this is actually a two-stage idealization. Firstly, actual outer codes do not have the infinite length required to achieve capacity. Secondly, we will see that achieving capacity requires that the inputs to the effective channel have a complex Gaussian distribution. This is an idealization of the actual transmitter, where the effective channel inputs are often QAM modulated.

Assuming the idealized outer code, one can obtain the capacity of the effective channel. This capacity depends on both the space-time inner code and the underlying MIMO fading channel. In particular, it can be used as a metric to compare different space-

time codes operating over the same MIMO fading channel. Also, it can be used as an optimization metric to design space-time inner codes. The advantage of this capacity-based approach is its simplicity. We will see that unlike the error probability of the supercode, information-theoretic analysis of the effective channel is quite simple. However, because of the implicit idealization of the outer code, the capacity-based approach does not accurately measure the performance for a finite-length, non-ideal outer code.

7.4 Organization of the Following Chapters

In the remainder of this work, we will use the capacity-based approach to obtain broad design rules for space-time inner codes. Note that the capacity of the effective channel depends not only on the space-time inner code, but also on the MIMO fading channel.

In the next chapter, we will review the rich literature on the information theoretic analysis of Rayleigh fading MIMO channel alone. Because of fading nature of the channel, it will prove necessary to introduce a relatively new kind of capacity, namely outage capacity.

In Chapter 9, we will extend the analysis to the space-time coded effective channel. This extension is simple because, as discussed in Section 2.4.1, the effective channel for a linear space-time code resembles the MIMO fading channel itself.

Combining the results of these two analyses, we obtain broad design rules for space-time inner codes in Chapter 10.

CHAPTER 8

Information Theoretic Analysis of MIMO Static Fading Channels

In this chapter, we provide an overview of information theoretic analysis of static fading channels. Much of the discussion is drawn from the vast literature on the topic, notably [6][4][5], although some of the proofs are our own. Specifically, the notion of outage probability and outage capacity of fading channels will be introduced. A high-SNR analysis of these quantities yields respectively the diversity order and multiplexing order of fading channels. These asymptotic quantities reflect the diversity and multiplexing abilities of fading channels, and will be used to analyze space-time inner codes in the next chapter. In this chapter, we also discuss the multiplexing-diversity trade-off curve [36].

8.1 Outage Probability and Diversity Order

In this section, we introduce the notion of outage, and discuss the outage probability and diversity order of MIMO fading channels. Recall, from (1), the quasi-static, linear MIMO fading channel model $\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k$. The data rate of transmission, say R_b bits / s / Hz, is given by the entropy rate of the channel input vector sequence $\mathbf{x}_k$. Recall that the signal-to-noise energy ratio is given by $S = \mathrm{E}[\|\mathbf{H}\mathbf{x}_k\|^2]/N_0$. Thus, the data rate and energy fix the entropy and average energy of the transmit vectors $\{\mathbf{x}_k\}$. Given these, an optimal encoder design should shape the distribution of the random vectors $\{\mathbf{x}_k\}$ so as to maximize

the mutual information between the channel input and output vectors. For a wide range of SNRs, the optimum distribution[1] [4] is that the elements of $\mathbf{x}_k$ are mutually independent zero-mean, complex Gaussian random variables with the same variance. For such an input distribution, the mutual information at an SNR S and a channel matrix $\mathbf{H}$ is given by [4]

$$I(S, \mathbf{H}) = \log_2 \det\left(\mathbf{I}_r + \frac{S}{t}\mathbf{H}\mathbf{H}^*\right) \text{ b / s / Hz.} \tag{105}$$

$I(S, \mathbf{H})$ is the *available capacity* corresponding to the channel matrix $\mathbf{H}$, namely the highest data rate that can be transmitted while maintaining zero error probability. Since the channel matrix $\mathbf{H}$ is random, so is the available capacity.

If the transmitter knew the channel, it could adapt the data rate R_b to be equal to the available capacity $I(S, \mathbf{H})$. However, in this work, we are concerned with the case where the transmitter does not know the channel. Consequently, it has to pick some data rate R_b without the guarantee of successful transmission. If $I(S, \mathbf{H}) \geq R_b$, the error probability can be made arbitrarily small with a well-chosen code. On the other hand, if $I(S, \mathbf{H}) < R_b$, the error probability cannot be made zero by any code. This event is called an *outage*. The occurrence of an outage does not necessarily imply that an error will occur, only that the error probability is non-zero. Thus, the probability of outage, called *outage probability*, provides an upper bound on the lowest achievable error probability.

By definition, the outage probability is $\Pr[I(S, \mathbf{H}) < R_b]$. In terms of the distribution function of the random available capacity $I(S, \mathbf{H})$ for an SNR S, namely

$$F(S, x) = \Pr[I(S, \mathbf{H}) < x], \tag{106}$$

1. Actually, at this stage, the distribution optimization problem is ill-posed, since the transmitter does not know the channel. We will return to this issue in the next chapter, while discussing Telatar's conjecture.

the outage probability is clearly equal to $F(S, R_b)$. For every value of the SNR S, we get a different distribution function[1] $F(S, x)$ depending on the statistical nature of the random channel matrix $\mathbf{H}$. For a wide variety of fading channels, the function $F(S, x)$ can be derived analytically. However, in this work, we are not interested in the exact expressions for $F(S, x)$, but only in its general behavior. In particular, we are interested in the behavior of outage probability at high SNR, which we now discuss.

8.1.1 The Diversity Order of MIMO Channels

As the SNR S increases, it is clear from (105) that the available capacity $I(S, \mathbf{H})$ increases for every channel matrix $\mathbf{H}$. Consequently, for any data rate R_b, the outage probability $F(S, R_b)$ decreases as the SNR increases. The *diversity order* quantifies the rate of this decrease at high SNR. Formally, it is defined as

$$\delta(R_b) = -\lim_{S \to \infty} \frac{\log F(S, R_b)}{\log S}. \tag{107}$$

Intuitively, the diversity order is the asymptotic slope of a log-log plot of outage probability vs. SNR. Clearly, a high diversity order represents a more rapid decrease of error probability with SNR, as is desirable.

To get some intuition into the diversity order, consider the outage probability $F(S, R_b)$ as a function of the SNR S, for a given value of R_b. Suppose it is infinitely differentiable in S. Then, one can expand $F(S, R_b)$ in a Laurent series in the powers of S, namely

$$F(S, R_b) = \frac{f_i(R_b)}{S^i} + \frac{f_{i+1}(R_b)}{S^{i+1}} + \ldots. \tag{108}$$

---

1. Some textbooks define the distribution function of a random variable X as $\Pr[\mathrm{X} \leq x]$, instead of $\Pr[\mathrm{X} < x]$.We consistently use the latter definition.

Note that the coefficients $f_i(R_b)$ of the expansion (108) are functions of the data rate R_b. Also, the coefficients corresponding to positive powers of S are all zero, since $F(S, R_b)$ uniformly decreases as S increases. As S goes to infinity, the term with the highest power of S dominates. Substituting $F(S, R_b) \approx f_i(R_b)/S^i$ in (107), we see that the diversity order is equal to i, namely the least power of $(1/S)$ that has a non-zero coefficient in the Laurent series expansion of $F(S, R_b)$.

From the above discussion, whenever the Laurent series expansion of $F(S, R_b)$ exists and is known in closed form, it can be used to compute the diversity order. For example, consider a 1-input, 1-output (scalar) Rayleigh fading channel. It is easy to show that the outage probability at SNR S is $F_{1,1}(S, R_b) = 1 - \exp(-(2^{R_b} - 1)/S)$. The first non-zero term in the Laurent series expansion is $(2^{R_b} - 1)/S$, implying that the diversity order of a 1-input, 1-output (scalar) Rayleigh fading channel is $\delta(1, 1) = 1$. Note that the diversity order is independent of the data rate R_b, as long as $0 < R_b < \infty$.

The extension of the above result to general MIMO Rayleigh fading channels is more complicated. The distribution function $F_{t,r}(S, x)$ of the available capacity of a t-input, r-output Rayleigh fading channel is known [51]. However, its Laurent series expansion is not known in closed form, and is therefore not useful in obtaining the diversity order. Instead, one can directly compute the limit in (107) using the clever analysis of [36][51]. We do not reproduce the well-known steps here, but merely state the final result below.

Theorem 2. The diversity order $\delta(t, r)$ of a t-input, r-output Rayleigh fading channel is equal to tr for all finite, non-zero data rates.

Note that the generalization $\delta(t, r) = tr$ agrees with the special case $\delta(1, 1) = 1$ derived above. Intuitively, this result is satisfying, since it confirms the intuitive notion of the

diversity benefit of multiple antennas. With t transmit and r receive antennas, there are tr independent Rayleigh fading links, and we expect to get tr-times the fade resistance of any one link, namely $\delta(1, 1) = 1$. Also note that $\delta(t, r) = tr$ agrees with the full raw diversity order of a general space-time code derived in Chapter 3. This is to be expected. Both the diversity order and full raw diversity order measure diversity with the best possible code. The only difference is that the diversity order holds for continuous Gaussian-distributed inputs, while the raw diversity order assumes a discrete input alphabet. The fact that the two agree shows that the difference in input distributions is not significant.

8.2 Outage Capacity and Multiplexing Order

The outage probability is an upper bound on the minimum possible error probability for a fading channel, at a fixed data rate. The outage capacity does the reverse, namely it finds the maximum possible data rate for a fixed outage probability. More precisely, for a given SNR S, as the data rate R_b increases, the outage probability $F(S, R_b) = \Pr[I(S, \mathbf{H}) < R_b]$ increases. The *outage capacity* is defined as the maximum data rate at which the outage probability $F(S, R_b)$ is less than some target value p_o. In other words, the outage capacity for an SNR S and target outage probability p_o is given by

$$C(S, p_o) = \sup\{R_b : F(S, R_b) < p_o\}. \tag{109}$$

Note that for any non-zero data rate, the outage probability $F(S, R_b)$ is strictly greater than zero. Consequently, the Shannon capacity, which is defined as the maximum data rate which guarantees zero error probability is zero for fading channels. Outage capacity replaces Shannon capacity as a measure of the data-carrying ability of fading channels.

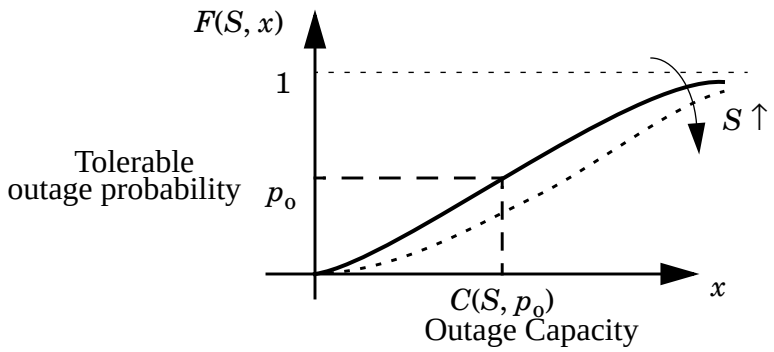

Fig. 9. Sketch to illustrate calculation of outage probability and outage capacity.

Instead of requiring zero error probability, a tolerance limit of p_o on the outage probability is placed, and the corresponding maximum data rate is measured.

Pictorially, the outage capacity is just computed by looking up the value of x such that $F(S, x) = p_o$, as shown in Fig. 9. As SNR increases, we mentioned that the distribution function $F(S, x)$ decreases pointwise for all x. As shown graphically in the sketch of Fig. 9, the distribution function moves down to a new position shown by the dotted curve. Consequently, it is easy to see for the same value of p_o, the outage capacity $C(S, p_o)$ increases with the SNR S. We now quantify the rate of this increase at high SNR.

8.2.1 The Multiplexing Order of MIMO Fading Channels

The *multiplexing order* of a fading channel is defined as

$$\mu(p_o) = \lim_{S \to \infty} \frac{C(S, p_o)}{\log S}. \tag{110}$$

Intuitively, it is the asymptotic slope of a plot of outage capacity vs. log SNR. In other words, at high SNR, every doubling of SNR increases the outage capacity by $\mu(p_o)$ bits/s/Hz. Clearly, a high multiplexing order represents a rapid increase of outage capacity with SNR, and is therefore desirable.

In the remainder of this section, we will show that, for a wide class of t-input, r-output fading channels, the multiplexing order is equal to $\min(t, r)$, irrespective of the target outage probability p_o[1]. An intuitive understanding of this result is obtained by analyzing the dependence of available capacity $I(S, \mathbf{H}) = \log \det(\mathbf{I}_r + (S/t)\, \mathbf{H}\mathbf{H}^*)$ on SNR S. Note that the random channel matrix $\mathbf{H}$ has dimension $r \times t$. If it has full rank of $m = \min(t, r)$, the matrix $\mathbf{H}\mathbf{H}^*$ has exactly m non-zero singular values, say $\lambda_1, \lambda_2, \ldots, \lambda_m$. Writing the determinant of $(\mathbf{I}_r + (S/t)\, \mathbf{H}\mathbf{H}^*)$ in terms of these, we get the equivalent expression

$$I(S, \mathbf{H}) = \log \prod_{i=1}^{m} \left(1 + \frac{S}{t}\lambda_i\right). \tag{111}$$

for available capacity. When the SNR S goes to ∞, the '1 + ' above can be neglected in comparison to the second term, which is linear in S. Thus, at high SNR

$$I(S, \mathbf{H}) \approx \log S^m \prod_{i=1}^{m} \left(\frac{\lambda_i}{t}\right) = m \log S + \log \prod_{i=1}^{m} \left(\frac{\lambda_i}{t}\right). \tag{112}$$

From (112), the available capacity is roughly the sum $m \log S$ and a constant term, independent of S. Therefore, it increases as $m \log S$ for all full-rank channel matrices $\mathbf{H}$. One intuitively expects that the outage capacity behaves similarly to the available capacity, implying that multiplexing order is $m = \min(t, r)$. This is made more precise in the following theorem [52], which has not been proved earlier, to the best of our knowledge.

Theorem 3. If the channel matrix $\mathbf{H}$ of a t-input, r-output fading channel has full rank with probability one, then the multiplexing order $\mu(p_o)$ equals $m = \min(t, r)$ for all $0 < p_o < 1$.

---

1. Though the multiplexing order does not depend on p_o, the outage capacity clearly does. We will discuss the influence of p_o on outage capacity at high SNR, in Section 8.3.

Before proving the theorem, we point out one crucial difference between the diversity and multiplexing orders. We saw that the diversity order depends strongly on the fading statistics of the channel, and is consequently difficult to compute in general. In contrast, the multiplexing order is easily computed to be $\min(t, r)$ for a wide class of fading channels, using Theorem 3. In particular, Rayleigh fading channels are known [4] to have full rank with probability one, resulting in the following corollary.

Corollary 2. The multiplexing order of a t-input, r-output fading channel is equal to $\min(t, r)$.

8.2.2 Proof that the Multiplexing Order is $\min(t, r)$

In this section, we prove Theorem 3. We first define

$$X(S, p_o) = \frac{2^{C(S, p_o)}}{S^m}, \text{ giving} \tag{113}$$

$$C(S, p_o) = m \log S + \log X(S, p_o). \tag{114}$$

Dividing by $\log S$ and taking limit as the $S \to \infty$, the multiplexing order (110) reduces to

$$\mu(p_o) = m + \lim_{S \to \infty} \frac{\log X(S, p_o)}{\log S}. \tag{115}$$

In order to show $\mu(p_o) = m$, it suffices to show the second limit above vanishes, or equivalently, that $X(S, p_o)$ is finite and bounded away from zero as $S \to \infty$. We first derive a more convenient expression for $X(S, p_o)$. Substituting $C(S, p_o) = \sup\{R_b : F(S, R_b) < p_o\}$ from (109) in the definition of $X(S, p_o)$, and using the fact that 2^x is a strictly increasing function, we get

$$X(S, p_o) = \sup\{x: \Pr\left[\frac{2^{I(S, \mathbf{H})}}{S^m} < x\right] < p_o\}. \quad (116)$$

It will prove useful to explicitly define $f(S, \mathbf{H}) = \frac{2^{I(S, \mathbf{H})}}{S^m}$. Substituting (111) for $I(S, \mathbf{H})$,

$$f(S, \mathbf{H}) = \frac{2^{I(S, \mathbf{H})}}{S^m} = \prod_{i=1}^{m} \left(\frac{\lambda_i}{t} + \frac{1}{S}\right). \quad (117)$$

Defining $g(S, x) = \Pr[f(S, \mathbf{H}) < x]$, (116) can be rewritten as

$$X(S, p_o) = \sup\{x: g(S, x) < p_o\}. \quad (118)$$

First, we remark that $X(S, p_o)$ is a non-decreasing function of the SNR S. This follows from the fact that $S_1 > S_2 \Rightarrow f(S_1, \mathbf{H}) < f(S_2, \mathbf{H}) \Rightarrow g(S_1, x) \geq g(S_2, x) \Rightarrow X(S_1, p_o) \leq X(S_2, p_o)$. Now, we can prove the following lemmas, bounding $X(S, p_o)$ from above and below.

Lemma 6. For all $p_o < 1$ and all SNR $S > 1$, $X(S, p_o) \leq X(1, p_o) < \infty$.

Proof: Since $X(S, p_o)$ is a non-decreasing function of the S, we have $X(S, p_o) \leq X(1, p_o)$ for all SNR $S > 1$. It remains to prove $X(1, p_o) < \infty$. The random variable $f(1, \mathbf{H})$ is a real-valued transformation of the random elements of $\mathbf{H}$, and is therefore a well-behaved random variable with no point masses at infinity. For all well-behaved random variables, the distribution function approaches one as $x \to \infty$ [53]. Thus,

$$\lim_{x \to \infty} g(1, x) = \lim_{x \to \infty} \Pr[f(1, \mathbf{H}) < x] = 1. \quad (119)$$

The limiting value $g(1, x) = 1$ can be arbitrarily closely approached by increasing x. More precisely, for all $p_o < 1$, there is an $X_o < \infty$ such that $g(1, x) > p_o$ for all $x > X_o$. From (118), this implies $X(1, p_o) \leq X_o < \infty$, which proves the lemma. The condition $S > 1$ was chosen arbitrarily in the lemma. Any non-zero value can be chosen as the lower limit for S, since we are only interested in a finite upper bound for $X(S, p_o)$.

Lemma 7. If **H** is full-rank with probability one, for all $p_o > 0$, there is a corresponding $x_o > 0$ such that $X(S, p_o) \geq x_o$ for all SNR S.

Proof: It is easy to see that $g(S, x)$ is upper-bounded, according to

$$g(S, x) = \Pr\left[\prod_{i=1}^{m}\left(\frac{\lambda_i}{t} + \frac{1}{S}\right) < x\right] \leq \Pr\left[\prod_{i=1}^{m}\frac{\lambda_i}{t} < x\right]. \tag{120}$$

The latter quantity is obtained by substituting $S = \infty$ in the definition of $f(S, \mathbf{H})$ in (117). Therefore, we will denote it as $g(\infty, x)$. As $x \to 0$, the limit of $g(\infty, x)$ is

$$\lim_{x \to 0} g(\infty, x) = \lim_{x \to 0} \Pr\left[\prod_{i=1}^{m}\frac{\lambda_i}{t} < x\right] \tag{121}$$

$$= \Pr\left[\prod_{i=1}^{m}\frac{\lambda_i}{t} \leq 0\right]. \tag{122}$$

Equation (122) is the probability that at least one of the singular values λ_i is zero, or that **H** is not full rank, which is zero by hypothesis. So, as $x \to 0$, $g(\infty, x) \to 0$. Given any $p_o > 0$, there is an $x_0 > 0$ such that $g(\infty, x) < p_0$ for all $x < x_0$. Using $g(S, x) \leq g(\infty, x) < p_0$ for all $x < x_0$ in (118), we get $X(S, p_o) \geq x_o > 0$. This proves the lemma.

Proof of Theorem 3: Lemma 6 says that $X(1, p_o) < \infty$. Also, if **H** is full-rank with probability one, Lemma 7 shows $0 < x_o \leq X(S, p_o)$. Combining the two, we get $0 < x_o \leq X(S, p_o) < X(1, p_o) < \infty$. Substituting the two bounds in (115), the limit involving $X(S, p_o)$ is bounded above and below by zero. Thus, the second limit vanishes and we get $\mu(p_o) = m = \min(t, r)$, proving the theorem.

8.3 The Outage Capacity Asymptote

In non-fading channels, one can achieve zero error probability by operating below the non-zero Shannon capacity. However, the outage phenomenon in fading channels results in a fundamental trade-off between data rate, error probability and SNR. The outage probability and outage capacity are obtained by fixing one of the two variables, namely the data rate and error probability respectively. Correspondingly, their asymptotic variation with SNR yields the diversity and multiplexing orders, which quantify the diversity and multiplexing benefits of MIMO channels. This leads one to seek an information theoretic metric that measures diversity and multiplexing benefits simultaneously. In this section, we argue that the outage capacity asymptote is one such metric. In the next section, we will briefly discuss the more comprehensive multiplexing-diversity trade-off curve [36].

From (114), we have the expression $C(S, p_o) = m \log S + \log X(S, p_o)$ for the outage capacity. As $S \to \infty$, the outage capacity approaches its *asymptote*

$$\overline{C}(S, p_o) = m \log S + \alpha(p_o), \tag{123}$$

where $\alpha(p_o)$ is given by the limit

$$\alpha(p_o) = \lim_{S \to \infty} \log X(S, p_o). \tag{124}$$

Note that this limit is well-defined and finite, since $X(S, p_o)$ has a well-defined non-zero limit from Lemma 6 and Lemma 7. From (123), the plot of the asymptote $\overline{C}(S, p_o)$ vs. $\log S$ is a straight line, whose slope is equal to the multiplexing order m. In this section, we will focus on the *zero-offset* of the asymptote, namely $\alpha(p_o)$. In particular, we will show that it is higher for channels with higher diversity order. Thus, the outage capacity asymptote contains information about both the multiplexing and diversity gains of MIMO channels.

Intuitively, a lower value of the target outage probability p_o places a stricter constraint on the data rate, and reduces the outage capacity. Since $\alpha(p_o)$ is the only part of the asymptotic outage capacity that depends on p_o, it must decrease as p_o decreases. To confirm this, note that $X(S, p_o) = \sup\{x: g(S, x) < p_o\}$ decreases with p_o, for all SNR S. Consequently, the limit $\alpha(p_o)$ of $\log X(S, p_o)$ also decreases with p_o.

We will now argue that for channels with high diversity order, $\alpha(p_o)$ decreases faster as p_o decreases. To see this, consider the sketches in Fig. 10. The upper sketch shows the outage capacity asymptotes for outage probabilities p and $q < p$. Both the asymptotes have slope equal to the multiplexing order μ, but different zero offsets $\alpha(p)$ and $\alpha(q)$ respectively. The difference $\Delta y = \alpha(p) - \alpha(q)$ represents the outage capacity loss incurred by reducing the target outage probability from p to q. The difference Δx represents the SNR gap between the two capacity asymptotes. From the slope μ, we see that

$$\Delta y / \Delta x = \mu. \tag{125}$$

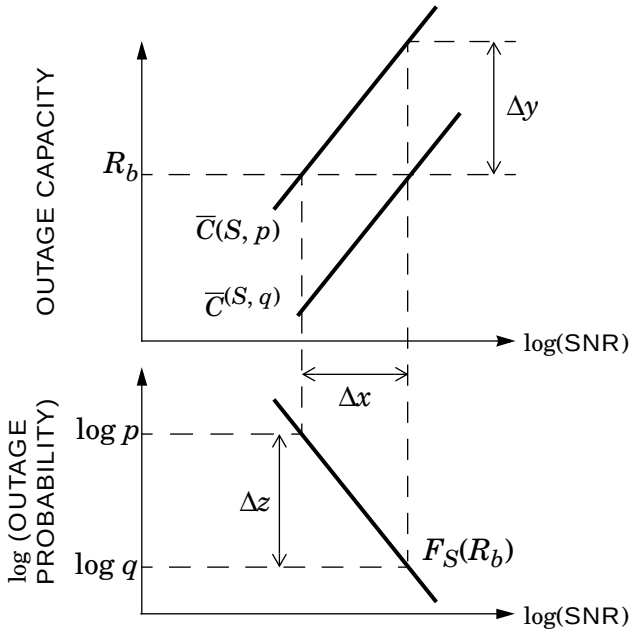

Fig. 10. Sketch of outage capacity and outage probability asymptotes.

We will now use the lower sketch to relate the SNR Δx to the diversity order δ. The lower sketch shows the asymptotic plot of log outage probability versus log SNR for a data rate R_b in the high SNR region. By definition, the slope of this asymptote is the diversity order δ. A reduction of the outage probability from p to q requires a drop along the y-axis of $\Delta z = \log(p) - \log(q)$. The corresponding increase in log SNR is equal to the SNR gap Δx. From the slope of the line, it is clear that

$$\Delta z / \Delta x = \delta. \tag{126}$$

Substituting $\Delta z = \log(p) - \log(q)$ in the above, we see that the SNR gap is

$$\Delta x = \frac{\log p - \log q}{\delta}. \tag{127}$$

From (127), the SNR gap Δx between outage capacity asymptotes for different outage probabilities is inversely proportional to the diversity order δ. Thus, suppose one plots the outage capacity asymptotes for different values of target outage probability. Then, the higher the diversity order of the channel, the closer these asymptotes are. Viewed differently, the higher the diversity order, the smaller the capacity loss incurred by decreasing the target outage probability. This follows by using (127) for Δx in (125) to get

$$\Delta y = \alpha(p) - \alpha(q) = \frac{\mu}{\delta}(\log p - \log q). \tag{128}$$

The above analysis also gives a method to obtain outage capacity asymptotes for different target outage probabilities. Suppose the zero offset $\alpha(p)$ of the outage capacity asymptote for the target outage probability p is known. Then, for all lower target outage probabilities $q < p$, (128) yields the following expression for the zero-offset $\alpha(q)$, namely

$$\alpha(q) = \frac{\mu}{\delta}\log q + \alpha(p) - \frac{\mu}{\delta}\log p. \tag{129}$$

It must be mentioned that the above expression is approximate. The actual capacity and diversity plots approach the asymptotic straight lines sketched in Fig. 10, but are not exactly straight lines themselves.

The approximation (129) shows qualitatively that the outage capacity is higher for channels with higher diversity order. Fixing p, the first term $(\mu / \delta) \log(q)$ in (129) dominates as the target outage probability q decreases. If two channels have the same capacity order μ but different diversity orders say $\delta_1 > \delta_2$, we have

$$\log(q)\,(\mu/\delta_1) > \log(q)\,(\mu/\delta_1), \tag{130}$$

since $\log(q) < 0$. Since this is the dominant term in the zero-offset, we see that the higher diversity channel has a greater zero-offset, and hence, a higher asymptotic outage capacity than the lower diversity channel, when q is sufficiently low.

In this section, we have shown that the zero-offset of the outage capacity asymptote carries diversity information, and is higher for channels with a higher diversity order. Unfortunately, the latter result is approximate, and holds only for low outage probabilities. Its main implication is that the outage capacity asymptote carries information about both the multiplexing and diversity gains of MIMO fading channels, in its slope and zero-offset respectively.

8.4 The Multiplexing-Diversity Trade-Off Curve

In the previous section, we obtained a unified multiplexing-diversity picture by studying the asymptotic outage capacity for different target outage probabilities. One can perform a similar approximate analysis of the outage probability asymptotes at different data rates. Similar conclusions are arrived at, namely the outage probability asymptotes at

different data rates are packed closer together for channels with higher multiplexing order. Also, channels with high multiplexing order have lower outage probabilities when the data rate is high enough. Again, these results are instructive, but somewhat imprecise.

One reason for the vagueness of the above results is that the different data rates for evaluating the outage probability asymptote are arbitrarily chosen. In [36] (see also [54]), an ingenious method of choosing the different data rates was suggested. Common sense, and the multiplexing order discussion, tell us that the achievable data rate increases in proportion to log SNR. Consequently, the different data rates are chosen not arbitrarily, but according to $R_b = g_{\mathrm{m}} \log S$, where the *multiplexing gain* g_{m} quantifies how much multiplexing we expect out of the MIMO channel. Now, the outage probability at SNR S is given by $F(S, g_{\mathrm{m}} \log S)$. Similar to the diversity order analysis, one can find the asymptotic slope of log outage probability vs. log SNR, yielding the *diversity gain*

$$g_{\mathrm{d}} = -\lim_{S \to \infty} \frac{\log F(S, g_{\mathrm{m}} \log S)}{\log S}. \tag{131}$$

Clearly, as the multiplexing gain g_{m} increases, the data rate $g_{\mathrm{m}} \log S$ for any SNR increases. Consequently, the outage probability $F(S, g_{\mathrm{m}} \log S)$ increases. Substituting in (131), we intuitively expect that the diversity gain g_{d} is a decreasing function of the multiplexing gain g_{m}. The plot of g_{d} vs. g_{m} gives a comprehensive picture of this decrease, and is called the multiplexing-diversity tradeoff curve.

Given a value of g_{m}, obtaining the corresponding value of g_{d} from (131) is a difficult task. The computation depends strongly on the variation of the outage probability $F(S, R_b)$ with SNR, which depends strongly on the channel characteristics. For the specific case of Rayleigh fading channels, the computation was performed in [36] to show the following.

Theorem 4. Consider a Rayleigh fading channel with t input and r outputs. The multiplexing-diversity trade-off curve is a piecewise linear function connecting the points $(k, (t-k)(r-k))$, $k = 0, 1, \ldots, \min(t, r)$.

In particular, Theorem 4 implies that for an integer value of the multiplexing gain g_m, the corresponding diversity gain is given by $(t - g_m)(r - g_m)$.

The advantage of the trade-off curve is that it is a comprehensive metric, measuring the multiplexing and diversity abilities of a fading channel at high SNR. In particular, it encompasses the multiplexing and diversity orders. In order to compute the diversity order, the data rate is kept fixed even as the SNR increases to infinity, hence the diversity order is merely the diversity gain g_d when the multiplexing gain g_m set to 0. To verify this, note that the diversity order tr is obtained for Rayleigh fading channels, both by directly finding diversity order in Theorem 2 and by substituting $g_m = 0$ in the expression $g_d = (t - g_m)(r - g_m)$ of Theorem 4. Similarly, it is easy to see that multiplexing order of fading channels is the multiplexing gain g_m corresponding to zero diversity gain. Again, for Rayleigh fading channels, this new formula gives the same multiplexing order $\min(t, r)$ as the direct computation in Theorem 3. In other words, the multiplexing and diversity orders are respectively the x- and y- intercepts of the trade-off curve.

The disadvantage of the trade-off curve is that it depends strongly on the fading characteristics, and is difficult to compute analytically. In particular, researchers have been unable to compute the trade-off curve for the effective MIMO channel formed by the combination of linear space-time codes and Rayleigh fading channels. Instead, we will use the simpler outage capacity asymptote to analyze the effective channel in the next chapter. The analysis will be used to obtain simple design rules for space-time inner codes.

CHAPTER 9

Information Theoretic Analysis of Space-Time Codes

In the last chapter, we discussed the information theoretic analysis of Rayleigh fading MIMO channels. In this chapter, we extend the analysis to the effective channel formed by the combination of the space-time code and the Rayleigh fading MIMO channel. We discuss the outage probability and outage capacity of the effective channel, and the effect of the rate and raw diversity order of a space-time code on its outage capacity asymptote. This analysis will be used in the next chapter to obtain design rules for space-time inner codes.

9.1 The Available Capacity of the Effective Channel

Consider a linear space-time inner code with K complex inputs in a block of length N, operating over a t-input, r-output Rayleigh fading channel. Let the $2Nt \times 2K$ real encoding matrix of the space-time code be $\mathbf{M}$. In Section 2.4.1, we discussed the effective channel formed by the combination of the linear space-time code and the Rayleigh fading MIMO channel. We obtained the input-output relation of the effective channel, namely

$$\hat{\mathbf{y}} = \hat{\mathbf{G}}\mathbf{M}\hat{\mathbf{u}} + \hat{\mathbf{n}}. \tag{132}$$

In what follows, we will be concerned with the transfer matrix $\hat{\mathbf{G}}\mathbf{M}$ of the effective channel matrix, which we call $\mathbf{H}_{\text{eff}}$ for convenience. In particular, we will study the effect

of $\mathbf{M}$ on the behavior of $\mathbf{H}_{\text{eff}}$. Recall from (11) that the $2Nr \times 2t$ matrix $\hat{\mathbf{G}}$ is obtained by applying the complex-to-real transformations (8) to the block diagonal matrix $\mathbf{G}$, which just contains N copies of the channel matrix $\mathbf{H}$ on its diagonal. Thus, $\mathbf{H}_{\text{eff}}$ depends not only on $\mathbf{M}$, but also on the random fading channel matrix $\mathbf{H}$ through $\hat{\mathbf{G}}$. Consequently, $\mathbf{H}_{\text{eff}}$ is also random.

As mentioned in Section 2.4.1, the effective channel strongly resembles the MIMO fading channel (1). The only differences are that the transfer $\mathbf{H}_{\text{eff}}$ is real, and has different dimensions, namely $2Nr \times 2K$, from the $r \times t$ complex MIMO channel matrix $\mathbf{H}$. After accounting for these minor differences, the information-theoretic analysis in the previous section extends to the effective channel. To begin, when the random fading channel matrix is $\mathbf{H}$, the available capacity of the effective channel at SNR S can be shown to be

$$J(S, \mathbf{M}, \mathbf{H}) = \frac{1}{2N} \log \det\left(\mathbf{I}_{2Nr} + \frac{2NS}{\|\mathbf{M}\|_{\mathcal{F}}^2} \hat{\mathbf{G}} \mathbf{M} \mathbf{M}^{\mathrm{T}} \hat{\mathbf{G}}^{\mathrm{T}} \right) \text{ b / s / Hz.} \tag{133}$$

For clarity, we have chosen the notation $J(S, \mathbf{M}, \mathbf{H})$ to explicitly separate the fading and encoding components of the effective channel matrix, $\mathbf{H}$ (or $\hat{\mathbf{G}}$) and $\mathbf{M}$ respectively. From the available capacity, one can define all the other quantities of the previous chapter. In particular, the distribution function

$$G(S, \mathbf{M}, x) = \Pr[J(S, \mathbf{M}, \mathbf{H}) < x] \tag{134}$$

of the available capacity at SNR S is used to obtain the outage probability corresponding to a data rate R_b namely $G(S, \mathbf{M}, R_b) = \Pr[J(S, \mathbf{M}, \mathbf{H}) < R_b]$. Correspondingly, the outage capacity for a fixed target outage probability p_{o} is defined as

$$D(S, \mathbf{M}, p_{\text{o}}) = \sup\{R_b : G(S, \mathbf{M}, R_b) < p_{\text{o}}\}. \tag{135}$$

We will loosely call the outage probability and capacity of the effective channel as, respectively, the outage probability and capacity of the space-time code itself.

9.2 The Outage Probability and Diversity Order of Space-Time Codes

In this section, we focus on the outage probability of space-time inner codes, which gives an upper bound on the error probability with a capacity-achieving outer code. Thus, the outage probability is analogous to the union bound of Chapter 3, which served as an upper bound on the error probability when there is no outer code. In Chapter 5 and Chapter 6, we discussed search strategies to find encoding matrices that minimize the union bound. Extending the analogy, one possible approach to space-time inner code design is to use outage probability as optimization metric. The optimization problem is stated precisely below.

> **Outage Probability Optimization Problem**: The SNR S, data rate R_b, channel dimensions $r \times t$, channel fading statistics (for example, Rayleigh), and encoding matrix dimensions $2Nt \times 2K$ are given. Find the $2Nt \times 2K$ encoding matrix $\mathbf{M}$ that minimizes the outage probability $G(S, \mathbf{M}, R_b)$.

The outage probability optimization problem is open, primarily because of the difficulty in obtaining a general closed form expression for $G(S, \mathbf{M}, R_b)$. We do not attempt to solve the problem here. Instead, we discuss a conjecture by Telatar [4], which gives the lowest possible outage probability that can be achieved by any space-time code operating on a given Rayleigh channel. We will subsequently discuss the conjecture's interesting implications for the design of space-time inner codes.

9.2.1 Telatar's Conjecture and Its Implications

Consider the information flow diagram of any space-time code, shown in Fig. 11. The space-time code encodes its input vector $\mathbf{u}$ to obtain the composite transmit vector $\mathbf{x}$, which is sent across the MIMO fading channel, yielding the composite receive vector $\mathbf{y}$. For notational clarity, we will use script variables to indicate random variables along with their distributions. Thus, $\mathcal{U}$ represents the distribution of the random input vector $\mathbf{u}$. Note that the distribution $\mathcal{U}$ along with the encoding matrix $\mathbf{M}$ fixes the distribution $\mathcal{X}$ of the composite transmit vector $\mathbf{x}$. In particular, since the encoding matrix $\mathbf{M}$ is fixed, $\mathbf{x}$ is known given $\mathbf{u}$, and so the entropy $H(\mathcal{X} | \mathcal{U})$ is zero. This is useful in the following standard information theoretic manipulation [55], which expands the mutual information $I(\mathcal{U}, \mathcal{X}; \mathcal{Y} | \mathbf{H})$ in two different ways using the chain rule.

$$I(\mathcal{U}, \mathcal{X}; \mathcal{Y} | \mathbf{H}) = I(\mathcal{U}; \mathcal{Y} | \mathbf{H}) + I(\mathcal{X}; \mathcal{Y} | \mathcal{U}, \mathbf{H}) \quad (136)$$

$$= I(\mathcal{X}; \mathcal{Y} | \mathbf{H}) + I(\mathcal{U}; \mathcal{Y} | \mathcal{X}, \mathbf{H}) \quad (137)$$

Now, $I(\mathcal{U}; \mathcal{Y} | \mathcal{X}, \mathbf{H}) = 0$ because the Markovian nature of the chain in Fig. 11 ensures that $\mathcal{Y}$ is conditionally independent of $\mathcal{U}$ given the intermediate variable $\mathcal{X}$. Also,

$$I(\mathcal{X}; \mathcal{Y} | \mathcal{U}, \mathbf{H}) \leq H(\mathcal{X} | \mathcal{U}, \mathbf{H}) \leq H(\mathcal{X} | \mathcal{U}). \quad (138)$$

We already saw that $H(\mathcal{X} / \mathcal{U})$ is zero, implying $I(\mathcal{X}; \mathcal{Y} | \mathcal{U}, \mathbf{H}) = 0$. Thus, the second terms in the right hand sides of (136) and (137) are zero, giving

$$I(\mathcal{U}; \mathcal{Y} | \mathbf{H}) = I(\mathcal{X}; \mathcal{Y} | \mathbf{H}). \quad (139)$$

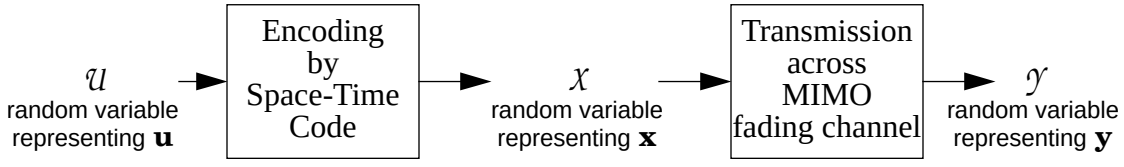

Fig. 11. Information flow diagram for space-time coding.

Now, for a given fading channel matrix **H**, the available capacity of the space-time code is equal to $(1/N)$[1] $I(\mathcal{U}; \mathcal{Y} | \mathbf{H})$ b / s / Hz. The corresponding outage probability at a data rate R_b is $\Pr[(1/N)\, I(\mathcal{U}; \mathcal{Y} | \mathbf{H}) < R_b]$. Substituting (139), we see that the outage probability at a data rate R_b is equal to

$$\Pr[(1/N)\, I(\mathcal{X}; \mathcal{Y} | \mathbf{H}) < R_b]. \tag{140}$$

Note that the distribution $\mathcal{X}$ determines the distribution of $\mathcal{Y}$ given **H**, and hence the mutual information $(\mathcal{U}; \mathcal{Y} | \mathbf{H})$. Thus, the distribution $\mathcal{X}$ determines the outage probability (140). Now, the encoding process places a constraint on the distribution $\mathcal{X}$, since **x** has to be the output of a linear space-time code with some encoding matrix **M**. If one relaxed this constraint, and allowed some arbitrary distribution $\mathcal{X}$, the best possible distribution yields a lower bound on the outage probability of any space-time code. Telatar's conjecture [4] gives the optimum (outage probability minimizing) distribution $\mathcal{X}$.

Conjecture 2. The outage probability (140) is minimized by a distribution $\mathcal{X}^*$ where the N individual transmit vectors $\{\mathbf{x}_i\}$ constituting the composite transmit vector **x** have identical zero-mean Gaussian distributions, and are independent of each other. The $t \times t$ covariance matrix Φ of each transmit vector is of the form $(N_0 S/k)$ $\mathbf{D}_k$, where $\mathbf{D}_k$ is the diagonal matrix

$$\mathbf{D}_k = \mathrm{diag}(\underbrace{1, 1, \ldots, 1}_{k \text{ 1s}}, \underbrace{0, 0, \ldots, 0}_{t-k \text{ 0s}}) \,. \tag{141}$$

1. The factor $(1/N)$ accounts for the fact that the signaling interval on the effective channel is N times as long as that on the MIMO fading channel.

The number of ones, k, should be chosen to optimize the outage probability. Note that the scaling (N_0S/k) ensures that the total transmit energy is equal to N_0S, which is consistent with an SNR S and noise energy N_0.

The conjectured optimum distribution $\mathcal{X}^*$ has independently distributed input vectors for different signaling intervals across the fading channel. It uses only a well-chosen number k of the inputs of the fading channel, and treats them identically. Note that the corresponding outage probability is (140)

$$\Pr[(1/N)\, \mathrm{I}(\mathcal{X}^*; \mathcal{Y}/\mathbf{H}) < R_b] = \Pr[\log_2 \det\left(\mathbf{I}_r + \frac{S}{k}\mathbf{H}\mathbf{D}_k\mathbf{H}^*\right) < R_b]. \tag{142}$$

Since this is conjectured to be the best possible distribution $\mathcal{X}^*$, (142) gives the conjectured value of the lowest outage probability of any space-time code with any input distribution. In particular, we get the conjectured lower bound

$$G(S, \mathbf{M}, R_b) \geq \min_k \Pr[\log_2 \det\left(\mathbf{I}_r + \frac{S}{k}\mathbf{H}\mathbf{D}_k\mathbf{H}^*\right) < R_b] \tag{143}$$

on how far the outage probability can be reduced. This is a powerful bound, since it is independent of the codelength N and the number of code inputs K. The optimal number of active inputs k varies with the data rate. For a wide range of data rates, the optimal number of active channels k is equal to t. However, for low data rates, the optimal value of k is lower than t [4].

9.2.2 Is the Serial to Parallel Converter Optimal?

Consider the serial to parallel converter as a space-time code. As described in Section 2.5.1, it has a length $N = 1$, takes in $K = t$ complex inputs and transmits them one on each available input to the t-input, r-output MIMO channel. Thus, the effective transmit vector $\mathbf{x}$ is equal to the input vector $\mathbf{u}$, and the input-output relation for the effective channel is $\mathbf{y} = \mathbf{Hu} + \mathbf{n}$, which is identical to that of the MIMO channel. If we chose the distribution of $\mathbf{u}$ to be $(N_0 S/k)\,\mathbf{D}_k$ for the optimally chosen k, it is easy to see that the S/P converter achieves exactly the conjectured optimum outage probability in (143). Thus, if Telatar's conjecture is true (it has so far not been proved or disproved, to our knowledge) one can achieve optimum outage probability for any given rate by choosing the S/P converter as space-time code, and optimally choosing the number of ON antennas.

The optimality of the S/P converter can also be motivated from another viewpoint. The famous data processing theorem of information theory [55] says that any form of coding can at best achieve the fundamental information-carrying capacity of the channel. The data processing theorem is not directly useful here because the Shannon capacity of the channel with or without coding is zero. However, the basic philosophy of the data processing theorem is that any kind of coding can only impair the information theoretic limits of communication. Since the S/P converter does effectively no coding, the effective channel is the same as the MIMO channel. Consequently, the S/P converter is expected to be optimum in terms of outage probability and other information theoretic quantities.

The conjecture that the S/P converter is optimum poses an existential question for space-time inner codes, i.e., it questions the need for any space-time inner code more sophisticated than the simple S/P converter. We will return to this issue in Chapter 10.

While discussing Telatar's conjecture, we have briefly touched upon the problem of optimizing the input distribution to a fading channel, with the objective of minimizing the outage probability. This is, by its own right, an important open problem (recall the discussion of the analogous open problem in Section 5.6.1 for the case of stand-alone space-time codes). For the remainder of this work, we will return to the earlier assumption that all input symbols are independent and have an identical complex Gaussian distribution, for all the channels involved. For the specific case of the S/P converter, the uniform distribution corresponds to Telatar's conjectured optimum distribution (141) with the number of active inputs k set to t. As stated earlier, the optimum k is indeed equal to t for high data rates and high SNR. For all such SNR, the lower bound (143) becomes

$$G(S, \mathbf{M}, R_b) \geq F(S, R_b), \tag{144}$$

where $F(S, R_b)$ is the outage probability of the MIMO fading channel, discussed in Section 9.2 of the previous chapter.

9.2.3 The Achievable Diversity Order of Space-Time Codes

We now proceed to analyze the outage probability at high SNR. The diversity order of the effective channel quantifies the asymptotic speed with which the outage probability $G(S, \mathbf{M}, R_b)$ decreases with SNR S. It is defined as

$$\chi(R_b) = -\lim_{S \to \infty} \frac{\log G(S, \mathbf{M}, R_b)}{\log S}. \tag{145}$$

Note that $\chi(R_b)$ is the diversity order of the space-time code corresponding to $\mathbf{M}$ with the best possible outer code, hence we call it *achievable diversity order* of the space-time code. At high SNR, the bound $G(S, \mathbf{M}, R_b) \geq F(S, R_b)$ from (144) holds, giving

$$\chi(R_b) \leq -\lim_{S \to \infty} \frac{\log F(S, R_b)}{\log S}. \tag{146}$$

The right hand side of (146) is just the diversity order *tr* of the Rayleigh fading channel from Theorem 2 of the previous chapter. Thus, with the best possible outer code, a space-time can achieve the diversity order of the Rayleigh fading channel. When this happens, the space-time code is said to have full achievable diversity order. In particular, note that the S/P converter has the same effective channel as the underlying MIMO channel itself, hence its outage probability is equal to $F(S, R_b)$. Consequently, the S/P converter in fact has full achievable diversity order *tr*. This is remarkable, since we saw in Section 2.5.1 that the raw diversity order of the S/P converter is only *r*. This trend in the S/P converter is part of a general rule, stated below.

> **Remark 6.** Any space-time code has two diversity-measuring quantities. The *raw* diversity order measures the diversity order the space-time code with uncoded inputs from a discrete alphabet. It is a lower bound to the *achievable* diversity order, namely the maximum diversity order achievable by the use of a well-chosen outer code and (possibly continuous) input alphabet.

While the raw diversity order of a space-time code is easily calculated using the rank rule, calculating the achievable diversity order for a general space-time code is an open problem. This is not surprising, considering the difficulty in calculating the diversity order of a general MIMO fading channel, as discussed in Section 8.1.

One case where the achievable diversity order can be calculated is when the space-time code already has full raw diversity order tr. The achievable diversity order cannot be any less, so it has got to be tr as well. Examples of such codes are the Alamouti code for $t = 2$, and the LCF codes (see Section 2.5) for any number of transmit antennas.

For many other space-time codes, the achievable diversity order can be obtained by inspection of the effective channel. One example of this is the S/P converter, where we noted that the effective channel is the same as the MIMO fading channel, and hence, the achievable diversity order is equal to the full diversity order. Another example is the repetition code, discussed below.

> **Example 1.** In every signaling interval across a t-input, r-output Rayleigh fading channel, the repetition code takes in $K = 1$ complex input and repeats it on all t channel inputs after scaling by $(1/\sqrt{t})$ to preserve the energy. Thus, it has a length $N = 1$. It is clearly a strictly linear space-time code whose encoding matrix has dimension $t \times 1$, and equals $\mathbf{M}_{\text{rep}} = (1/\sqrt{t})[1, 1, \ldots, 1]^{\text{T}}$. The effective channel is an $r \times 1$ channel given by $\mathbf{H}_{\text{rep}} = \mathbf{H}\mathbf{M}_{\text{rep}}$. Each element of the effective channel is the scaled sum of i. i. d. complex Gaussians, which can easily be shown to also be an identical complex Gaussian random variable. Thus, the effective channel $\mathbf{H}_{\text{rep}}$ is statistically equivalent to an 1-input, r-output Rayleigh fading channel. From this, we see that the diversity order of the effective channel, which is equal to the achievable diversity order of the repetition code, is equal to r.

The repetition code is a pathological example. All the other space-time codes that we have mentioned in this work can be shown to have full achievable diversity order.

Rate and raw diversity order were two important factors in determining the performance of stand-alone space-time codes. However, the S/P converter shows that even with a low raw diversity order, a space-time code can achieve full diversity as an inner code. Thus, high raw diversity order is not crucial for a space-time code to be a good inner code. On the other hand, analysis of outage capacity in the next section will show that high rate is crucial for a space-time inner code.

9.3 Multiplexing Order of Space-Time Codes Cannot Exceed the Rate

In this section, we analyze the outage capacity (135) of space-time codes at high SNR. Specifically, we focus on the multiplexing order of space-time codes, defined as

$$\nu(\mathbf{M}, p_o) = \lim_{S \to \infty} \frac{D(S, \mathbf{M}, p_o)}{\log S}. \tag{147}$$

As discussed in the previous chapter, the multiplexing order represents the slope of the outage capacity vs. log SNR asymptote. The analysis of multiplexing order of space-time codes closely follows the derivation of the multiplexing order of MIMO fading channels in Section 8.2. One only needs to account for the statistical differences between the effective channel matrix and the Rayleigh fading MIMO channels.

To begin, we write out the available capacity $J(S, \mathbf{M}, \mathbf{H})$ in terms of the singular values of the effective channel matrix $\mathbf{H}_{\text{eff}} = \hat{\mathbf{G}}\mathbf{M}$. Now, $\hat{\mathbf{G}}$ has dimension $2Nr \times 2Nt$, hence its rank cannot exceed its minimum dimension, namely $\min(2Nr, 2Nt) = 2N\min(r, t)$. Similarly, the rank of the $2Nt \times 2K$ matrix $\mathbf{M}$ is at most $2N\min(t, K/N)$. Now, the rank of the product $\mathbf{H}_{\text{eff}} = \hat{\mathbf{G}}\mathbf{M}$ cannot exceed the ranks of either $\hat{\mathbf{G}}$ or $\mathbf{M}$. Combining the above bounds, and noting that K/N is by definition the rate R of the space-time code, we get

$$\text{rank}(\mathbf{H}_{\text{eff}}) \leq 2Nn, \text{ where} \tag{148}$$

$$n = \min(t, r, R). \tag{149}$$

The matrix $\mathbf{H}_{\text{eff}}\mathbf{H}_{\text{eff}}^{\text{T}}$ in the available capacity expression (133) has dimension $2Nr \times 2Nr$. Let $\kappa_1 \geq \kappa_2 \geq \ldots \geq \kappa_{2Nr} \geq 0$ be its ordered singular values. From the rank bound (148), at most the first $2Nn$ of these singular values are non-zero. Rewriting the determinant in (133) in terms of the non-zero singular values, we get the equivalent expression

$$J(S, \mathbf{M}, \mathbf{H}) = \frac{1}{2N} \log \prod_{i=1}^{2Nn} \left(1 + \frac{2NS}{\|\mathbf{M}\|_{\mathcal{F}}^2} \kappa_i\right) \tag{150}$$

for the available capacity. Note the similarity between this expression and the corresponding expression (111) for the available capacity of the Rayleigh fading channel $\mathbf{H}$ alone. In particular, one can use the same intuitive argument used there, to estimate the multiplexing order of the effective channel. Suppose a specific instance of the channel matrix $\mathbf{H}_{\text{eff}}$ has rank ρ. Then, only the first ρ terms of the product in (150) remain. At high SNR, one can neglect the '1 + ' in each term, and approximate the available capacity by

$$\frac{1}{2N} \log S^{2Nn} \prod_{i=1}^{\rho} \left(\frac{2N}{\|\mathbf{M}\|_{\mathcal{F}}^2} \kappa_i\right) = \frac{\rho}{2N} \log S + \frac{1}{2N} \log \prod_{i=1}^{\rho} \left(\frac{2N}{\|\mathbf{M}\|_{\mathcal{F}}^2} \kappa_i\right). \tag{151}$$

The approximation (151) grows as $(\rho / 2N)\log S$ indicating that the multiplexing order of the effective channel is $(\rho / 2N)$. From (148), we know that the rank ρ is at most $2Nn$, implying that the multiplexing order is at most n. In order to meet this upper bound, we need to ensure that the 'typical' effective matrix has full rank. This imprecise argument is now stated precisely in the following theorem.

Theorem 5. The multiplexing order of a rate-R linear space-time code, when operating over a t-input, r-output fading channel satisfies

$$\nu(S, \mathbf{M}, p_o) \le n = \min(t, r, R). \tag{152}$$

If the effective channel matrix $\mathbf{H}_{\text{eff}} = \hat{\mathbf{G}}\,\mathbf{M}$ has full rank with probability one, then the upper bound is achieved, giving a multiplexing order $\nu(S, \mathbf{M}, p_o) = n$.

Proof: The proof follows exactly along the lines of the proof of Theorem 3 in Section 8.2.2. In particular, analogous to $X(S, p_o)$ defined in (113), we define

$$Y(S, \mathbf{M}, p_o) = \frac{2^{2ND(S, \mathbf{M}, p_o)}}{S^{2Nn}}, \text{ giving} \tag{153}$$

$$D(S, \mathbf{M}, p_o) = n \log S + \frac{1}{2N} \log Y(S, p_o). \tag{154}$$

Dividing by $\log S$ and taking limit as S goes to infinity, we get

$$\nu(S, \mathbf{M}, p_o) = n + \frac{1}{2N} \lim_{S \to \infty} \frac{\log Y(S, \mathbf{M}, p_o)}{\log S}. \tag{155}$$

First, following the proof of Lemma 6, it is easy to prove that $Y(S, p_o) \le Y(1, p_o) < \infty$ for all $p_o < 1$. Thus, the limit involving $Y(S, p_o)$ on the right hand side of (155) is at most zero, implying $\nu(S, \mathbf{M}, p_o) \le n$. This proves the first part of the theorem.

The proof of the second part follows the analogue of Lemma 7. Thus, if the effective channel matrix is full rank with probability one, one can adapt the proof of Lemma 7 to show that $Y(S, p_o) > y_o > 0$. Again, substituting in (155), the limit involving $Y(S, p_o)$ is indeed equal to zero. This proves that, if the effective channel matrix is full rank with probability one, the multiplexing order is n, as claimed in the theorem.

According to Theorem 5, the rate of a space-time code places an upper bound on the multiplexing order. Recall from Section 2.3.1 that the rate of a space-time code is equal to the number of symbols multiplexed by the effective channel per signaling interval. Thus, the claim of Theorem 5 that the multiplexing order cannot exceed the rate is intuitively easy to understand. However, the condition to achieve that upper bound, namely that the effective channel matrix $\mathbf{H}_{\text{eff}} = \hat{\mathbf{G}}\mathbf{M}$ have full rank with probability one, is not direct and tractable. In particular, it is not clear how the encoding matrix $\mathbf{M}$ should be chosen in order to satisfy this condition. In the next section, we will assume a Rayleigh fading channel $\mathbf{H}$, and investigate how the encoding matrix $\mathbf{M}$ should be chosen to ensure a full-rank $\mathbf{H}_{\text{eff}}$.

9.4 Choosing an Encoding Matrix to Maximize Multiplexing Order

Suppose an encoding matrix $\mathbf{M}$ is chosen. Now, define the *bad channel set* $\mathcal{B}(\mathbf{M})$ to be the set of all $\mathbf{H}$ which gives a rank-deficient effective channel matrix, i.e.,

$$\mathcal{B}(\mathbf{M}) = \{\mathbf{H}\text{: } \hat{\mathbf{G}}\mathbf{M} \text{ is not full rank}\}. \tag{156}$$

Note that $\mathbf{H}_{\text{eff}} = \hat{\mathbf{G}}\mathbf{M}$ is not full rank if and only if the random channel matrix $\mathbf{H}$ belongs to the bad channel set $\mathcal{B}(\mathbf{M})$. Therefore, $\Pr(\mathbf{H}_{\text{eff}} \text{ is full rank}) = \Pr(\mathbf{H} \notin \mathcal{B}(\mathbf{M}))$. If these probabilities are one, then Theorem 5 assures us that the multiplexing order upper bound is achieved (152). In this section, we seek conditions that the encoding matrix $\mathbf{M}$ should satisfy in order to ensure $\Pr[\mathbf{H} \notin \mathcal{B}(\mathbf{M})] = 1$, or equivalently $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$.[1]

1. In measure-theoretic terms, we seek conditions on $\mathbf{M}$ to ensure that the corresponding bad channel set $\mathcal{B}(\mathbf{M})$ is a set of zero measure in the set of all Rayleigh fading channel matrices $\mathbf{H}$.

We will assume that the space-time code has rate R less than or equal to the full rate $\min(t, r)$. In particular, note that this means $R \leq r$, and consequently $2NR \leq 2Nr$. Since the rate is defined to be $R = K/N$, we get $2NR = 2K \leq 2Nr$. Thus, the assumption that $R \leq \min(t, r)$ implies that the $2Nr \times 2K$ effective channel matrix $\mathbf{H}_{\text{eff}}$ has at most as many columns as rows. Consequently, the rank of $\mathbf{H}_{\text{eff}}$ is equal to its column rank. In other words, $\mathbf{H}_{\text{eff}} = \hat{\mathbf{G}}\mathbf{M}$ does not have full rank if and only if its columns are linearly dependent, i.e., there is some non-zero *right nulling vector* $\hat{\mathbf{u}} \in \mathcal{R}^{2K}$ such that $\hat{\mathbf{G}}\mathbf{M}\hat{\mathbf{u}} = \mathbf{0}$. Thus, the bad channel set (156) can be written as

$$\mathcal{B}(\mathbf{M}) = \{\mathbf{H}: \exists \hat{\mathbf{u}} \in \mathcal{R}^{2K}, \text{ such that } \hat{\mathbf{u}} \neq \mathbf{0}, \text{ and } \hat{\mathbf{G}}\mathbf{M}\hat{\mathbf{u}} = \mathbf{0}\}. \tag{157}$$

We will now use this representation to derive conditions to ensure $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$.

Lemma 8. To achieve $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$, it is necessary to ensure that the encoding matrix $\mathbf{M}$ has a full rank.

Proof: It is easy to see that the $2Nt \times 2K$ $\mathbf{M}$ also has at least as many rows as columns. Its rank is equal to its column rank. In particular, if it does not have full rank, it has some non-zero right nulling vector $\mathbf{u}_{\mathbf{M}}$ such that $\mathbf{M}\mathbf{u}_{\mathbf{M}} = \mathbf{0}$. But this implies $\hat{\mathbf{G}}\mathbf{M}\mathbf{u}_{\mathbf{M}} = \mathbf{0}$ irrespective of $\hat{\mathbf{G}}$. Substituting in (157), every possible instance of the channel matrix $\mathbf{H}$ belongs to the bad channel set, giving $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 1$. We have just shown that $\mathbf{M}$ not full rank $\Rightarrow \Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 1$. Reversing the argument, $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$ only if $\mathbf{M}$ has full rank, proving the lemma.

Lemma 8 provides a necessary condition on the encoding matrix $\mathbf{M}$. In order to obtain sufficient conditions, it will prove useful to split the analysis into two different cases, depending on the relation between the number of inputs and outputs of the fading channel.

9.4.1 Sufficient Conditions for the Case $t \leq r$

Lemma 9. Suppose $t \leq r$. Then, to achieve $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$, it is sufficient to ensure that the encoding matrix $\mathbf{M}$ has full rank.

Proof: Consider some matrix $\mathbf{H} \in \mathcal{B}(\mathbf{M})$. By definition, $\exists \hat{\mathbf{u}} \in \mathcal{R}^{2K}$, $\hat{\mathbf{u}} \neq \mathbf{0}$, $\hat{\mathbf{G}}\mathbf{M}\hat{\mathbf{u}} = \mathbf{0}$. First, we will use the assumption that $\mathbf{M}$ is full rank, and hence $\hat{\mathbf{u}} \neq \mathbf{0} \Rightarrow \mathbf{M}\hat{\mathbf{u}} \neq \mathbf{0}$. Thus, $\mathbf{H} \in \mathcal{B}(\mathbf{M})$ only if $\hat{\mathbf{G}}$ has a non-zero right nulling vector $\mathbf{M}\hat{\mathbf{u}}$, or equivalently, if $\hat{\mathbf{G}}$ does not have full column rank. Now, the second assumption $t \leq r$ implies that $\hat{\mathbf{G}}$ has at most as many rows as columns. Thus, the column rank of $\hat{\mathbf{G}}$ is also its overall rank. Combining the two arguments, $\mathbf{H} \in \mathcal{B}(\mathbf{M}) \Rightarrow \hat{\mathbf{G}}$ does not have full rank. It is easy to show [4] that $\hat{\mathbf{G}}$ does not have full rank if and only if $\mathbf{H}$ itself does not have full rank. Consequently, $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] \leq \Pr[\hat{\mathbf{G}}$ not full rank$] = \Pr[\mathbf{H}$ not full rank$]$. For Rayleigh fading matrices, the latter probability is known to be zero [4]. This implies $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$, proving the lemma.

Lemma 8 and Lemma 9 prove the following theorem, giving a simple necessary and sufficient condition for achieving the multiplexing order upper bound of Theorem 5.

Theorem 6. Let $t \leq r$. A linear space-time code with rate $R \leq t$ operating over a t-input, r-output Rayleigh[1] fading channel achieves the multiplexing order upper bound $n = \min(t, r, R) = R$ if and only if the encoding matrix $\mathbf{M}$ has full rank.

1. The restriction to Rayleigh fading channels is only for convenience. Lemma 8 holds irrespective of the fading channel statistics. Lemma 9 only requires that the channel matrix have full rank with probability one.

All practical linear space-time codes employ full-rank encoding matrices in order to guarantee unique decodability of the input signal. More precisely, the space-time code produces the composite transmit vector $\hat{\mathbf{x}} = \mathbf{M}\hat{\mathbf{u}}$. If $\mathbf{M}$ is not full rank, the encoding process is linearly irreversible, i.e., one cannot just perform a pseudo-inverse to get $\hat{\mathbf{u}}$ uniquely from $\hat{\mathbf{x}}$. Thus, a non-full rank $\mathbf{M}$ complicates decoding with no corresponding benefit, and is therefore never used.

Theorem 6 states that merely using a full-rank encoding matrix $\mathbf{M}$, for which there are other sound reasons as described above, ensures that the multiplexing order upper bound, equal to the rate, is achieved. Thus, when $t \leq r$, we can think of the rate of a space-time code as its multiplexing order.

9.4.2 Sufficient Conditions for the Case $t > r$

In the case $t > r$, a full rank encoding matrix is not sufficient to achieve the multiplexing order upper bound. Mathematically, the proof of Lemma 9 does not hold because it assumes $t \leq r$. Intuitively, here is an example of a full rank encoding matrix that does not achieve the multiplexing order upper bound.

Example 2. Let the Rayleigh fading channel have $t = 2$ inputs and $r = 1$ output. Consider a space-time code which takes in $K = 2$ complex inputs, say u_1 and u_2, and produces 2×1 transmit vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ for $N = 2$ signaling intervals, according to

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ 0 \\ 0 \end{bmatrix}. \tag{158}$$

Note that this is a (strictly) linear space-time code of rate $R = 2/2$ $(K/N) = 1$. The multiplexing order upper bound of Theorem 5 is $\min(2, 1, 1)$ $(\min(t, r, R))$, namely one. The actual multiplexing order achieved by the code can be obtained by inspection. This code essentially uses the Rayleigh fading MIMO channel once every two signaling intervals. Hence, it has exactly half the outage capacity of the 2-input, 1-output Rayleigh fading channel for all SNR. Thus, the actual multiplexing order of the space-time code is half, which is less than the upper bound one. However, it is easy to see that the encoding matrix here does indeed have full rank. (Intuitively, given $\mathbf{x}$, we can get $\mathbf{u}$, by just picking out the first two elements. So the encoding matrix has to be full rank.) This example shows that a full rank encoding matrix is not sufficient for achieving the upper bound.

Now, we will derive a new sufficient condition for the case $t > r$. Consider the N transmit vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ produced by the space-time code. Let $\mathcal{X}_i$ be the set of all values of the transmit vector $\mathbf{x}_i$. The linearity of the code imposes considerable structure on $\mathcal{X}_i$. In particular, for a strictly linear space-time code, it is easy to see that $\mathcal{X}_i$ is just a linear subspace of the set of all $t \times 1$ complex vectors. For general linear space-time codes, the space $\mathcal{X}_i$ itself is not linear, but the set of all transformed vectors $\hat{\mathbf{x}}_i$ is a linear subspace of the set of all $2t \times 1$ real vectors. One can interpret $\mathcal{X}_i$ as the column span of the rows of the encoding matrix $\mathbf{M}$ corresponding to the $\hat{\mathbf{x}}_i$. We now state a sufficient condition on the spaces $\{\mathcal{X}_i\}$, which guarantees that the multiplexing order upper bound is achieved.

Theorem 7. Let $t > r$. Consider a linear space-time code of length N with rate $R \leq r$ operating over a t-input, r-output Rayleigh fading channel. Then, in order to achieve the multiplexing order upper bound $n = \min(t, r, R) = R$, it is necessary to

use a full-rank encoding matrix $\mathbf{M}$. Further, it is sufficient (though not necessary) to ensure that each set $\mathcal{X}_i$ of all possible $\mathbf{x}_i$ is some non-zero subspace of a linear space $\mathcal{L}_i$ whose dimensions satisfies $1 \le l_i \le r$.

Proof: The fact that a full-rank encoding matrix is necessary follows from Lemma 8. Now, we assume that $\mathbf{M}$ is full rank and proceed to prove the sufficient condition. For any $\mathbf{H} \in \mathcal{B}(\mathbf{M}) \Leftrightarrow \hat{\mathbf{G}}\,\mathbf{M}\hat{\mathbf{u}} = \hat{\mathbf{G}}\,\hat{\mathbf{x}} = \mathbf{0}$ for some non-zero input $\hat{\mathbf{u}}$. Further, $\hat{\mathbf{G}}\,\hat{\mathbf{x}} = \mathbf{0}$ $\Leftrightarrow \mathbf{G}\mathbf{x} = \mathbf{0} \Leftrightarrow \mathbf{H}\mathbf{x}_i = \mathbf{0}$ for all the transmit vectors $\mathbf{x}_i$ corresponding to the composite vector $\mathbf{x}$. Note that since $\mathbf{M}$ is full rank and $\hat{\mathbf{u}} \neq \mathbf{0}$, $\hat{\mathbf{x}} \neq \mathbf{0}$, implying that at least one $\mathbf{x}_i$ $\neq \mathbf{0}$. Consequently, $\mathbf{H} \in \mathcal{B}(\mathbf{M}) \Rightarrow$ there is at least one $\mathbf{x}_i \neq \mathbf{0}$ such that $\mathbf{H}\mathbf{x}_i = \mathbf{0}$. Defining the *local bad channel set*

$$\mathcal{B}_i(\mathcal{X}_i) = \{\mathbf{H}\text{: } \mathbf{H}\mathbf{x}_i = \mathbf{0} \text{ for some } \mathbf{x}_i \in \mathcal{X}_i, \mathbf{x}_i \neq \mathbf{0}\}, \tag{159}$$

we see that $\mathbf{H} \in \mathcal{B}(\mathbf{M}) \Rightarrow \mathbf{H} \in \mathcal{B}_i(\mathcal{X}_i)$ for some i. Thus, $\mathcal{B}(\mathbf{M}) \subseteq \bigcup_{i=1}^{N} \mathcal{B}_i(\mathcal{X}_i)$. Hence, to show $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$, it suffices to show $\Pr[\mathbf{H} \in \mathcal{B}_i(\mathcal{X}_i)] = 0$ for all $i \in \{1, 2, \ldots, N\}$.

Now, by assumption $\mathcal{X}_i \subseteq \mathcal{L}_i$, a complex linear space of dimension l_i such that $1 \le l_i \le r$. Since $\mathcal{L}_i$ is a complex linear space, every vector in it can be written as the column span of a $t \times l_i$ complex matrix $\mathbf{B}_i$ with orthonormal columns. Every $\mathbf{x}_i \in \mathcal{X}_i$ is also in $\mathcal{L}_i$, so $\mathbf{x}_i = \mathbf{B}_i\mathbf{u}_i$ for some non-zero $\mathbf{u}_i$. Substituting in (159), the local bad set now is

$$\mathcal{B}_i(\mathcal{X}_i) = \{\mathbf{H}\text{: } \mathbf{H}\mathbf{B}_i\mathbf{u}_i = \mathbf{0} \text{ for some } \mathbf{u}_i \neq \mathbf{0}\}. \tag{160}$$

Define the transformed random matrix $\mathbf{H}_i' = \mathbf{H}\mathbf{B}_i$. Since $\mathbf{B}_i$ has orthonormal columns, it is easy to show [4] that $\mathbf{H}_i'$ is a Rayleigh fading matrix of dimension $r \times l_i$. Note that $\mathbf{H}\mathbf{B}_i\mathbf{u}_i = \mathbf{0} \Leftrightarrow \mathbf{H}_i'$ has linearly dependent columns $\Leftrightarrow \mathbf{H}_i'$ does not have full

rank, since $l_i \leq r$. Thus, $\Pr[\mathbf{H} \in \mathcal{B}_i(\mathcal{X}_i)] = \Pr[\mathbf{H}_i' \text{ not full rank}]$. Again using the fact Rayleigh fading matrices have full rank with probability one [4], the latter quantity is zero. We have just shown $\Pr[\mathbf{H} \in \mathcal{B}_i(\mathcal{X}_i)] = 0$ for all $i = 1, 2, \ldots, N$, implying $\Pr[\mathbf{H} \in \mathcal{B}(\mathbf{M})] = 0$. Thus, the condition for achieving the upper bound in Theorem 5 holds, completing the proof.

The sufficient condition presented above has an elegant intuitive interpretation. Note that the dimension of the set $\mathcal{X}_i$ represents the multiplexing rate of each individual transmit vector $\mathbf{x}_i$. Now, the rate R of the space-time code, which is the average of all these individual multiplexing rates, is by assumption less than or equal to the multiplexing order of the fading channel, namely $\min(t, r) = r$. The condition $1 \leq l_i \leq r$ requires that not only the average rate, but each individual multiplexing rate should also be less than or equal to the multiplexing order of the channel. Equivalently, the average multiplexing rate should be distributed roughly uniformly among all the individual transmit vectors. In particular, given a code which does not satisfy the sufficient condition, one can just redistribute the rate across the individual transmit vectors and obtain a new code that does satisfy the condition. Again, we come to the conclusion that it is easy to design a linear space-time code which achieves a multiplexing order equal to its rate, as promised by Theorem 5.

Example 3. We revisit Example 2 to illustrate the sufficiency condition derived in Theorem 7. Note that the two transmit vectors $\mathbf{x}_1 = \mathbf{u}$ and $\mathbf{x}_2 = \mathbf{0}$ of the code in Example 2 have dimensions 2 and 0 respectively. On the other hand, the condition in Theorem 7 requires that both transmit vectors have dimension one. (In this case, $r = 1$, so the upper and lower bounds on l_i are both one.) So, the code in Example 2 does not satisfy the sufficiency condition, therefore it is not surprising that it does not achieve the multiplexing order upper bound.

As promised in the discussion of Theorem 7, a simple rearrangement can satisfy the sufficient condition, and yield a code that achieves the multiplexing order upper bound. Consider the re-arranged code

$$\mathbf{x}' = \begin{bmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \end{bmatrix} = \begin{bmatrix} u_1 \\ 0 \\ u_2 \\ 0 \end{bmatrix}, \tag{161}$$

which is obtained by just swapping the second and third elements of the composite transmit vector $\mathbf{x}$ from (158). Now, both $\mathbf{x}_1'$ and $\mathbf{x}_2'$ have dimension one, and satisfy the sufficient condition of Theorem 7. Consequently, we expect the rearranged code to achieve a multiplexing order bound of $n = \min(2, 1, 1)$. This is easily verified by inspection. The rearranged code uses only the first input of the 2-input, 1-output Rayleigh fading channel (the second and fourth symbols of $\mathbf{x}'$, which correspond to the second input of the channel, are always zero). Thus, the effective channel is equivalent a 1-input, 1-output Rayleigh fading channel, with multiplexing order one.

Before closing this section, we emphasize that the dimensionality condition derived in Theorem 7 is only a sufficient, and not a necessary condition. In other words, codes whose spaces $\mathcal{X}_i$ do not satisfy the dimension constraint could still achieve a multiplexing order equal to the rate. For such codes, one can often tell by inspection whether or not the effective channel has full rank with probability one, as illustrated below.

Example 4. Consider the rate-one Alamouti code (Section 2.5.2) operating over a 2-input, 1-output Rayleigh fading channel. It is easy to see that the two transmit vectors of the Alamouti code belong to spaces of dimension 2. Therefore, the Alamouti code does not satisfy the sufficient condition of Theorem 7, namely that the spaces $\mathcal{X}_i$ have dimension less than or equal to $r = 1$. So, we are not sure if the Alamouti code achieves the multiplexing order upper bound. However, we can use the condition in Theorem 5 and try to find out if the effective channel is full rank with probability one. From Section 2.5.2, the effective channel, after some receiver signal processing, is given by $\mathbf{y}' = \|\mathbf{H}\|_{\mathcal{F}}^2\mathbf{u} + \mathbf{n}'$. In particular, the effective channel matrix is $\|\mathbf{H}\|_{\mathcal{F}}^2\mathbf{I}_2$, where $\mathbf{I}_2$ is the 2×2 identity matrix. For Rayleigh fading channels, $\|\mathbf{H}\|_{\mathcal{F}}^2$ is non-zero with probability one, implying that the effective channel $\|\mathbf{H}\|_{\mathcal{F}}^2\mathbf{I}_2$ is full rank with probability one. Then, we conclude that the Alamouti code achieves the multiplexing order upper bound of $n = \min(2, 1, 1) = 1$.

9.5 The Outage Capacity Asymptote

We have so far discussed the outage probability and outage capacity of space-time codes, and the corresponding asymptotic slopes, namely the achievable diversity order and the multiplexing order. In this section, we study the high-SNR outage capacity asymptote

for space-time codes. As discussed in Section 9.5, the outage capacity asymptote carries information about both the multiplexing and diversity aspects of fading channels. Following the discussion in Section 9.2.1, we see that the outage capacity of a space-time code cannot exceed that of the MIMO fading channel on which it operates. We aim to find out how close the outage capacity asymptote of the space-time code is to this upper limit.

By definition, the multiplexing order is equal to the slope of the outage capacity asymptote. A t-input, r-output Rayleigh fading channel has multiplexing order $\min(t, r)$. On the other hand, a space-time code with rate R less than the full rate $m = \min(t, r)$ has a multiplexing order of at most R. Thus, space-time codes with low rate have a shallower outage capacity asymptote than the fading channel. In particular, at high SNR, they achieve at most a fraction $R / \min(t, r)$ of the Rayleigh fading channel's outage capacity. Thus, low rate space-time codes suffer a dramatic capacity loss at high SNR.

As discussed in Section 9.5, low diversity order leads to a low zero offset of the outage capacity asymptote. Consequently, if the achievable diversity order of a space-time code is less than the diversity order tr of the Rayleigh fading channel, the code suffers a constant (offset) loss in the outage capacity asymptote. However, as mentioned earlier, even codes like the S/P converter with low raw diversity order have full achievable diversity order, implying that raw diversity order does not directly impact the outage capacity asymptote. Most practical space-time codes have full achievable diversity order, irrespective of their raw diversity order. There are codes which do not achieve full diversity order, and we will see one such code in Section 9.5.1 below. However, these codes are usually contrived and unnatural. We do not know of any practical space-time code that suffers an offset loss because of low achievable diversity order.

It is important to note the offset loss is only qualitatively related to the diversity order, but is not completely determined by it. Consequently, we can only say that codes with low achievable diversity order definitely suffer an offset loss. We cannot say the reverse, namely that codes with full achievable diversity order have zero offset loss.

9.5.1 Outage Capacity Asymptote: Illustrative Example

We now present an example to illustrate the effect of rate and raw diversity order on the outage capacity of space-time codes. We consider a Rayleigh fading channel with $t = 2$ inputs, and either one or two outputs. We compare two rate-one space-time codes: the Alamouti code, and the repetition code, discussed in Example 1. The 1% outage capacity (i.e., $p_o = 0.01$) is plotted vs. SNR in Fig. 12. The plots are obtained by generating many random channel matrices, and computing the corresponding available capacity using (105). Based on many such trials, a discrete approximation of the distribution function of the available capacity is generated, and used to compute the outage capacity, as discussed in Section 8.2.

Table 2: Multiplexing and Diversity Orders for $t = 2$ inputs, $r = 1$ outputs.

Code	Multiplexing Order	Reasoning	Achievable Diversity Order	Reasoning
No code (also S/P Converter)	1	$\min(t, r)$	2	tr
Alamouti (Rate $R = 1$)	1	Example 4	2	Raw diversity Order is also 2
Repetition Code (Rate $R = 1$)	1	$\min(t, r, R)$ Theorem 7	1	Example 1 Div. Order = r

Consider first the case of $r = 1$ output fading channel. The multiplexing and diversity orders of the fading channel, and the Alamouti and repetition codes are tabulated in Table 2. We see that both the Alamouti and repetition codes have the same multiplexing order as the 1-input, 1-output Rayleigh fading channel, namely one. This is verified by Fig. 12, where the asymptotic slopes of the Alamouti and repetition codes' capacity curves matches that of the fading channel. In addition to full multiplexing order, the Alamouti code also has full achievable diversity order, namely two. Due to its full diversity order, it is not expected to suffer a significant zero offset loss, when compared to the fading channel. Remarkably, as observed in Fig. 12 and proven in [16], the capacity penalty of the Alamouti code is zero when there is only one receive antenna. On the other hand, the repetition code has an achievable diversity order of one, which is a loss from the channel's diversity order of two. The lower diversity order results in a lower zero-offset of the capacity asymptote, and hence the constant capacity loss at high SNR seen in Fig. 12.

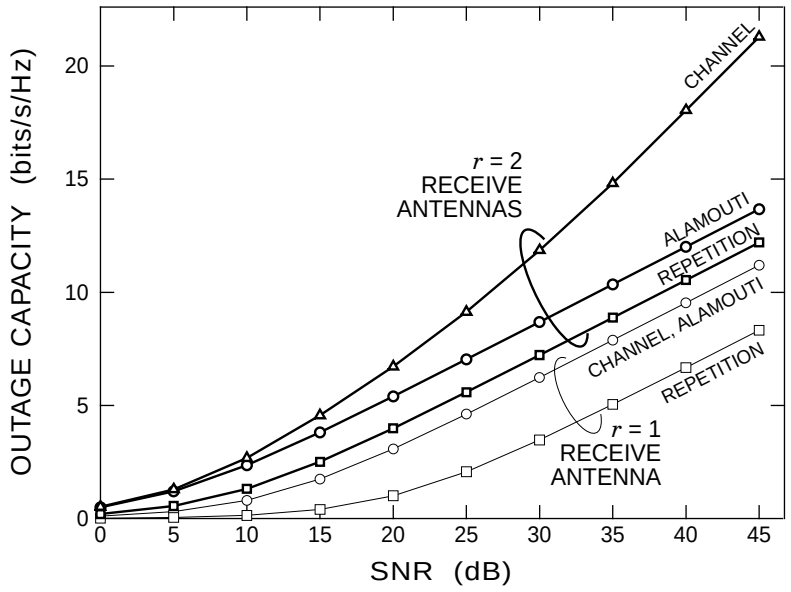

Fig. 12. Outage capacity versus SNR at 1% outage, assuming $t = 2$ transmit antennas.

Now, consider the case where the channel has $t = r = 2$ outputs. The multiplexing and diversity orders of the channel, and the Alamouti and repetition codes operating over the channel, are tabulated in Table 3. The channel has a multiplexing order of two, while the two rate-one codes have a multiplexing order of one. This agrees with Fig. 12, where the outage capacity curve corresponding to the underlying channel has a slope that is twice as steep as those corresponding to the two space-time codes. Consequently, both codes can achieve at most 50% of the outage capacity of the 2-input, 2-output fading channel at high SNR. The Alamouti code at least has full diversity order of four. The repetition code has diversity order two, which is less than the diversity order of the channel. Consequently, the repetition code suffers an additional offset loss when compared to the Alamouti code.

9.6 The Multiplexing-Diversity Trade-Off Curve: An Open Problem

In Section 8.4, we discussed the trade-off curve between multiplexing and diversity gains for a Rayleigh fading MIMO channel. In principle, one can extend the same definition (131) of diversity gain to the effective channel to obtain the diversity gain

$$h_{\mathrm{d}}(\mathbf{M}) = -\lim_{S \to \infty} \frac{\log G(S, \mathbf{M}, h_{\mathrm{m}} \log S)}{\log S}. \tag{162}$$

Table 3: Multiplexing and Diversity Orders for $t = 2$ inputs, $r = 2$ outputs.

Code	Multiplexing Order	Reasoning	Achievable Diversity Order	Reasoning
No code (also S/P Converter)	2	$\min(t, r) = 2$	4	$tr = 4$
Alamouti (Rate $R = 1$)	1	$\min(t, r, R)$ Theorem 6	4	Raw diversity Order is also 4
Repetition Code (Rate $R = 1$)	1	$\min(t, r, R)$ Theorem 6	2	Example 1 Div. Order = r

The problem is again that there is no simple closed form expression for the distribution function $G(S, \mathbf{M}, x)$ of the available capacity. Therefore, obtaining the multiplexing-diversity trade-off curve for a general space-time code remains an open problem. We have solved the problem partially here. By obtaining the multiplexing order (equal to $\min(t, r, R)$ for most codes), we have found the multiplexing gain for zero diversity gain. The achievable diversity, which is nearly always tr, is equal to the diversity gain for zero multiplexing gain. Thus, we have obtained two points on the trade-off curve, namely $(\min(t, r, R), 0)$ and $(0, tr)$.

For some special space-time codes, the entire trade-off curve can be obtained. For example, the S/P converter has the same trade-off curve as the t-input, r-output Rayleigh fading channel, since its effective channel is equal to the latter. Another example is the Alamouti code, which has the simple effective channel $\mathbf{y}' = \|\mathbf{H}\|_f^2 \mathbf{u} + \mathbf{n}'$. Since the multiplexing and diversity orders are one (because of rate one) and $2r$ respectively, the two known points on the curve are $(1, 0)$ and $(0, 2r)$. In [36], the effective channel of the Alamouti code was analyzed to show that the trade-off curve is in fact a straight line joining these points.

9.7 Summary

In this chapter, we have adapted the information theoretic analysis of Chapter 8 to the effective channel formed by the combination of a space-time code and MIMO fading channel. In particular, we pointed out that the raw diversity order of a space-time code is only a lower bound to the achievable diversity order, which is often equal to the full diversity order of the channel. On the other hand, we showed that the multiplexing order

of a space-time code is less than or equal to its rate. Most known linear space-time codes have a multiplexing order equal to rate. The implications of these results on the outage capacity asymptote were shown by discussion and example. In the next chapter, we will use the analysis derived here to derive broad design rules for space-time inner codes.

CHAPTER 10

Capacity-Based Design Rules for Space-Time Inner Codes

In the last chapter, we used information theory to analyze the maximum data rates and minimum error rates that can be achieved by supplementing a space-time inner code with the best possible outer code. In this chapter, we apply the results of that analysis to understand practical design issues for space-time inner codes.

10.1 The Importance of Using High Rate Inner Codes

We saw that the space-time codes with rate R less than the full rate $\min(t, r)$ have a low multiplexing order, and hence lose a significant fraction of the outage capacity at high SNR, when operating over a t-input, r-output Rayleigh fading channel. Consequently, to avoid capacity loss, it is clear that one must use space-time inner codes with full rate or more. (Rate being equal to full rate alone does not always guarantee full multiplexing order, but Theorem 6 and Theorem 7 show that most full-rate codes do have full multiplexing order.)

On the other hand, the raw diversity order is only a lower bound to achievable diversity order, and does not directly impact any of the information-theoretic quantities described in Chapter 9. Primarily, this is because a well-designed outer code can make up for the lack of transmit diversity in the space-time inner code. This is illustrated by the example of the S/P converter which has raw diversity order r, but full achievable diversity order of tr.

We conclude that in the presence of a powerful outer code, space-time inner codes must be designed to have full rate, but need not have high raw diversity order. The design rule is only a broad one. In particular, there is the issue of what precisely is a powerful outer code. Note that the capacity analysis assumes infinite length outer codes, which are designed to produce an optimal, continuous Gaussian distributed output. However, we now present simulation results which show that the results of capacity analysis hold even for sufficiently powerful binary codes like turbo [31] or LDPC [47][48] codes.

10.1.1 High-Rate Space-Time Inner Codes Are Better: Example

We compare two space-time inner codes operating over a 4-input 4-output Rayleigh-fading channel, assuming the outer code is a binary turbo code. In each codeword, the turbo code encodes 3200 input bits and produces 4800 bits per codeword, hence its rate is $2/3$. The turbo code has two parallel concatenated $\left[1, \frac{1+D+D^4}{1+D+D^2+D^3+D^4}\right]$ convolutional codes. The input bits are fed directly to the first convolutional code to obtain the first parity stream. On the other hand, the input bits are interleaved by a spread-20 interleaver before being fed to the second convolutional code, the parity outputs of which are deinterleaved to obtain the second parity stream. The parity streams from the two codes are punctured to achieve rate $2/3$. For the first parity stream, only the bits in positions 0, 4, 8, … are

retained. For the second parity stream, only the bits in positions 2, 6, 10, … are retained. After this puncturing, 4800 output bits remain. Following the bit-interleaved coded modulation strategy [56] (also [33]), these output bits are interleaved using a spread-24 interleaver, and Gray-mapped to complex QAM symbols for space-time encoding.

The space-time inner codes considered are the serial-to-parallel converter and an Alamouti-based general layered space-time code (GLST) [18]. The GLST code consists of two rate-one Alamouti codes operating in parallel over groups of two transmit antennas each. Its rate is twice that of an Alamouti code, namely two. Also, its length $N = 2$. It is easy to see that pairwise differences between GLST code matrices are full rank, and hence using the rank rule, the raw diversity order is $r\min(t, N) = 8$. In contrast, the S/P converter has higher (full) rate of $t = 4$, but lower raw diversity order of $r = 4$.

For a fair comparison, we fix the data rate of the two space-time inner codes. Since the GLST code has half the rate of the S/P converter, it has to use a higher constellation size to achieve the same data rate. In this case, the S/P converter uses 16-QAM modulation, while the GLST code uses 256-QAM modulation. Thus, both codes transmit 16 inputs bits per signaling interval. Scaling by the rate $2/3$ of the outer turbo code, the total data rate achieved by the concatenated architecture is 10.67 bits/s/Hz.

At this data rate, it takes $3200/10.67 = 300$ signaling intervals across the Rayleigh fading channel to transmit one frame, i.e., all the 4800 output bits of the turbo code. The Rayleigh fading channel is assumed to be constant in one frame, but varies independently from one frame to the next.

As an approximation to optimum decoding, the receiver does iterative decoding between the outer turbo decoder and a soft-output list sphere decoder for the inner space-time code [33] (see Chapter 12). Three turbo iterations are performed for each of the ten iterations between the outer turbo decoder and the inner space-time decoder. A frame error is said to occur when any of the 3200 input bits to the turbo code is incorrectly decoded. Fig. 13 shows a plot of the frame error rate vs. SNR. Each point represents a reading of at least 150 frame errors.

The multiplexing order of the S/P converter is equal to that of the 4-input, 4-output Rayleigh fading channel, namely $\min(4, 4) = 4$. In contrast, the multiplexing order of the GLST code is $\min(4, 4, 2) = 2$ from Theorem 6 (the GLST encoding matrix has full rank). We expect that the low multiplexing order of the latter code should lead to a loss of

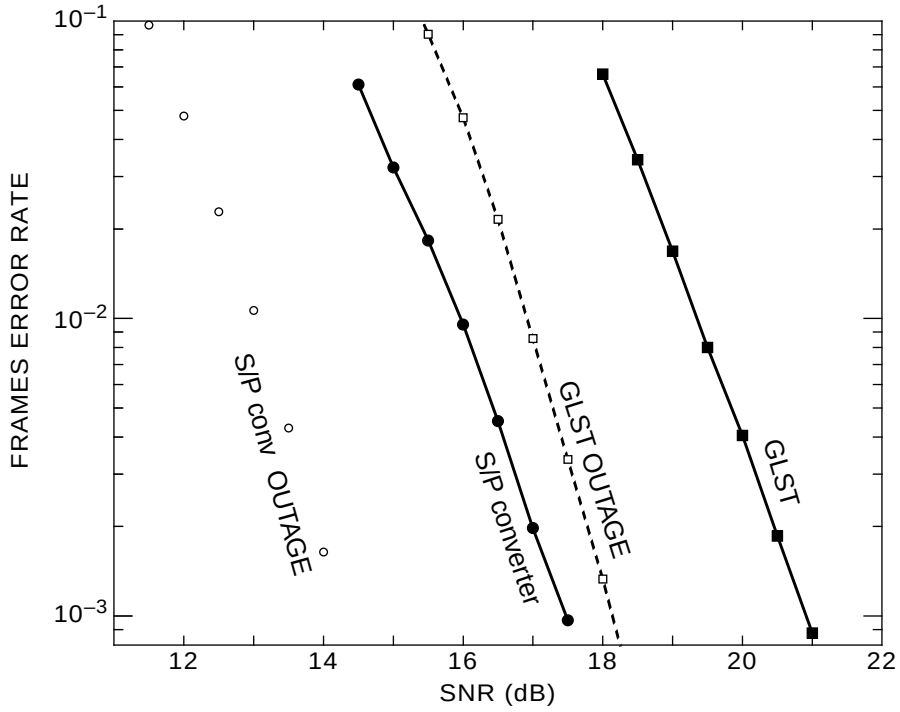

Fig. 13. Performance of S/P converter and GLST space-time inner codes with a turbo outer code, over a 4-input, 4-output Rayleigh fading channel at 10.67 bits/s/Hz.

capacity, and hence a loss of performance in the presence of the outer turbo code. This is confirmed by the plot, which shows the high-rate S/P converter outperforming the low-rate GLST encoder by nearly 3.5 dB, in spite of the latter code's higher raw diversity order.

For comparison, the outage probability is plotted against SNR, with the data rate fixed at 10.67 bits/s/Hz. Note that even with a code-length of just 4800 bits, the performance of the turbo code is within 3 dB of the outage probability curve at a word error rate of 10^{-3}. This indicates that the results of outage analysis hold even for binary outer codes with finite length. However, the actual word error rate curve with a turbo outer code is shallower than the outage probability curve for the S/P converter, indicating that the turbo code is still not strong enough to get the full diversity order of the channel. We conjecture that increasing the length of the turbo code will lead to full diversity order.

10.2 Is the Serial-to-Parallel Converter an Optimum Inner Code?

The simulation result in the last section showed in a specific example that the S/P converter outperforms the GLST code in the presence of a turbo outer code. In this section, we discuss the merits and demerits of the general transmitter architecture shown in Fig. 14. Coded bits from a binary outer code are interleaved and modulated to obtain complex input symbols to the S/P converter, which serves as the space-time inner code.

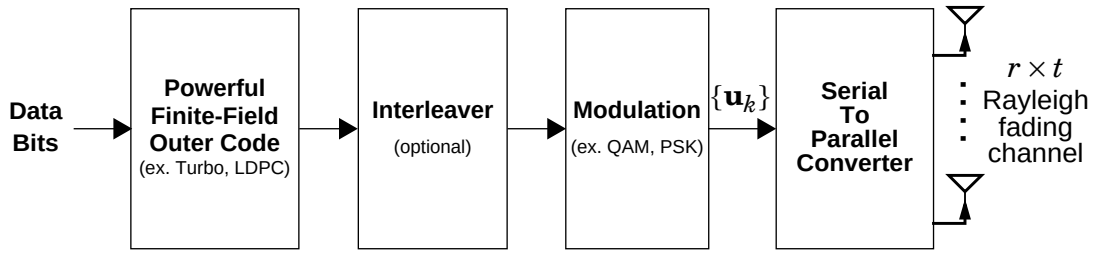

Fig. 14. Concatenation of a binary outer code with the S/P converter space-time code.

The architecture of Fig. 14 follows the same philosophy as the bit interleaved coded modulation architecture [56], which bootstraps codes designed for BPSK to obtain good performance for higher QAM modulation, hence avoiding the complicated design of trellis coded modulation techniques. Analogously, the architecture of Fig. 14 does away with sophisticated space-time inner coding, and exploits the fact that good binary codes are easy to design and decode using iterative techniques. Thus, one merit of the architecture is its simplicity. The question is whether this simplicity comes at the cost of optimality.

With an ideal infinite-length outer code and a Gaussian alphabet, Telatar's conjecture (see Section 9.2.2) implies that the S/P converter is an optimum inner code, i.e., it achieves least outage probability given data rate, and most data rate given outage probability. However, the practical concatenated architecture under consideration differs from this idea in two ways.

The first deviation is that the modulation alphabet is discrete (usually QAM). Arguably, this is not a very serious deviation. In other words, one might hope that even with a finite alphabet, the S/P converter is an optimum inner code provided the outer code has infinite length and can be optimally designed. We have no proof that this is the case, but we conjecture it is so.

However, the more serious deviation is that the outer code often has finite length, and belongs to a certain family, say turbo codes. In particular, the length of the outer code could seriously affect the performance of the concatenated transmitter. In the extreme case of no outer code (or effectively length one outer code), the diversity order achieved is the raw diversity order of the S/P converter, namely r. Given the actual length of the outer

code, it is an interesting open question to determine how close to optimum the S/P code is, as a space-time inner code. The family from which the outer code is drawn is also important. Since the outer code is the only source of transmit diversity, it must have a large minimum distance. Thus, codes with large minimum distance, like turbo and LDPC codes, are suitable outer codes in Fig. 14, but convolutional and Reed Solomon codes are unsuitable because of their low minimum distance.

10.3 The Alternative: Full-Rate, Full Raw Diversity Inner Codes

The S/P converter relies on the outer code to obtain diversity, and hence performs poorly for low outer code lengths. A more robust concatenated architecture would use a space-time inner code with full raw diversity order, and at least full rate. The raw diversity order ensures good performance for low outer code lengths, and full rate ensures high multiplexing order and consequently good performance for large outer code lengths. We saw in Chapter 4 that full rate, full raw diversity space-time codes are aplenty. One such inner space-time code can be used to replace the S/P converter in Fig. 14 in order to increase robustness.

The disadvantage of using a full rate, full diversity space-time code is the decoding complexity. In order to achieve full raw diversity, the inner space-time code must have length $N \geq t$. Also, the rate satisfies $R \geq \min(t, r)$. Consequently, the number of space-time code inputs per block satisfies

$$K = NR \geq t\min(t, r). \tag{163}$$

Recall from Section 2.4.1 that the effective channel of a linear space-time code has $2K$ inputs in one block, and produces $2Nr$ outputs. Given the $2Nr$ outputs, the receiver has to

produce hard or soft decisions about the $2K$ inputs in every space-time code block. We have assumed in this work that this is done optimally. While optimal decoding is simple in principle, its complexity increases exponentially with the number of inputs $2K$. Even with simplified decoding strategies like sphere decoding [25][33] (also see Chapter 11), the complexity increases in proportion to K^3 [57][58]. Thus, to ensure low complexity, it is desirable to keep K low. Attempting to achieve full diversity while simultaneously maintaining at least full rate places a lower bound (163) on the number of inputs, and hence on the complexity.

There is another subtle problem with the use of full diversity space-time codes as inner codes in a concatenated configuration. We have assumed that the outer code is designed independently of the inner code and the number of transmit antennas. In particular, let us say it is a binary turbo code, designed assuming independent BPSK transmission of the output bits over an AWGN channel. Instead, the bits are modulated into QAM symbols. Let each QAM symbol contain n_{Q} bits. Then, K such symbols, carrying Kn_{Q} bits are transmitted across the effective channel in one space-time code block. At the receiver, the soft decisions produced for all these Kn_{Q} bits are dependent. On the other hand, the iterative decoder for the turbo code works well only when the soft decisions are independent. For a larger value of K, the number of dependent bits Kn_{Q} is larger, and hence iterative decoding is more severely affected. Thus, for a long outer code, a full diversity space-time inner code might lead to poorer performance than the S/P converter.

10.4 Design of Concatenated Architectures: An Open Problem

In this chapter, we have discussed the concatenated architecture for space-time transmitters. Outage analysis suggests that space-time inner codes must have at least full rate, and leaves open the choice of raw diversity order. We have listed the relative merits and demerits of two possible choices of the inner code: the simple, but non-robust S/P converter with low raw transmit diversity order; and robust, but computationally demanding full raw diversity inner codes.

However, given constraints on the outer code length and computational complexity, it is an interesting open problem to design the best possible combination of outer code and inner space-time codes, namely the combination that minimizes error rate at a given data rate, or maximizes data rate given the acceptable error rate. This is one of the most important open problems in transmitter design for flat-fading MIMO channels.

This chapter concludes our discussion of transmitter design for MIMO channels. We now move on to receiver design.

CHAPTER 11

Tree-Pruning Detectors For MIMO Channels

In this work, we have so far focused on transmitter design for MIMO fading channels. We now turn our attention to the design of MIMO receivers. In contrast to transmitter design, the choice of optimality metrics is not a significant issue in receiver design. Given the transmitter structure, the data rate is automatically fixed. Consequently, the task of receiver design is merely to ensure minimize error probability. Further, the structure of the optimum receiver, namely the one that achieves minimum error probability, is often obvious. However, the optimum receiver is often computation-intensive. In this chapter and the next one, we present efficient algorithms to implement optimum or near-optimum receivers.

Two transmitter architectures for MIMO channels have been discussed in this work: a transmitter with a stand-alone linear space-time code; and the concatenation of an outer error correction code with an inner linear space-time code. As one would expect, different receiver structures need to be employed for each of these transmitters. In this chapter, we discuss the design of receivers for stand-alone linear space-time codes. In the next chapter, receiver design for concatenated transmitters will be discussed.

As stated in Chapter 3, the input to a stand-alone space-time code is typically drawn from a discrete alphabet, independently from block to block. Given the received signals in each space-time code block, the receiver employs a detection algorithm or *detector* to estimate the space-time code input in that block. The popular successive cancellation (SC) detector has low computational complexity, but leads to a high probability of estimation error. It is well known [7] that the detector that minimizes the probability of estimation error is the maximum likelihood (ML) detector. This fact was stated in Chapter 3, while deriving the union bound. Further, in Chapter 6, the error rates for various space-time codes with ML decoding at the receiver were shown. In this chapter, we present the ML detection algorithm that was used to obtain the simulation results of Chapter 6.

Graphically, the ML detection problem can be interpreted as the search for the cheapest leaf node in tree [7]. Sequential decoders like the Fano decoder and ZJ stack decoder (see [59] for a survey) for convolutional codes are also based on a search for the cheapest leaf node in a tree. Some attempts have been made to exploit this similarity, and adapt sequential decoders to obtain efficient MIMO detection algorithms (see, for example [60]). However, there are significant differences between the two problems. For instance, the Fano branch metric (see, for example [2]) used in sequential decoding does not naturally extend to the MIMO detection tree. More significantly, trees in MIMO detection have much smaller depths compared to those in the sequential decoding problem.

Another approach to MIMO detection is to employ lattice search algorithms developed in the computer science literature [24][28][29]. This approach yields the well-known sphere decoder [25-30], which efficiently implements ML detection for MIMO channels, when the transmit symbols belong to an integer lattice.

In this chapter, we reinterpret the sphere decoder as essentially a tree-search algorithm, that seeks the cheapest leaf node in the detection tree. We review a simple strategy for developing efficient tree-search algorithms, which was suggested in a homework problem of [7], and later extended in [23]. This basic strategy can be implemented in multiple ways, yielding a class of *tree-pruning* algorithms. We show that the sphere decoder belongs to this broad class of algorithms. We also derive a new tree-pruning algorithm called the *hybrid decoder*, which requires higher memory than the sphere decoder, but lends itself to high-speed parallel implementation. By placing limits on the memory available to the hybrid decoder, one obtains the *bounded stack* hybrid decoder, which allows one to control the worst-case computational complexity of the detection process, at the cost of increasing error rate.

11.1 A Precise Statement of the Detection Problem

We begin the discussion by precisely stating the ML detection problem. We consider a general linear memoryless M-input, P-output channel, whose input-output relation is

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\,\tilde{\mathbf{x}} + \tilde{\mathbf{n}}\,. \tag{164}$$

Note that the model (164) fits both the wireless fading channel (1) and the effective channel (12) for a linear space-time code. The algorithms presented here work for any channel of the form (164), but for illustrative simulation results, $\tilde{\mathbf{H}}$ will be assumed to be a $P \times M$ Rayleigh fading channel. In keeping with the rest of this work, the transmitter does not know $\tilde{\mathbf{H}}$, but the receiver knows it accurately. The elements of the noise vector $\tilde{\mathbf{n}}$ are independent, zero-mean complex Gaussian random variables of variance N_0.

The transmit alphabet or channel-input alphabet $\mathcal{X}$, namely the set of all possible input vectors $\tilde{\mathbf{x}}$, is assumed to be finite. For convenience of presentation, we assume that each element of $\tilde{\mathbf{x}}$ is drawn independently from a finite alphabet $\mathcal{A}$, known as the *symbol alphabet*. For example, $\mathcal{A}$ could be a PAM, QAM or PSK alphabet. (Note that the symbols could be real or complex.) Thus, $\mathcal{X}$ is equivalent to $\mathcal{A}^M$.

The linear MIMO channel (164) distorts the input vector $\tilde{\mathbf{x}}$ by causing different symbols to interfere at the receiver. This distortion is similar to inter-symbol interference (ISI) in wideband single-input, single-output (SISO) channels [23]. To combat ISI, SISO receivers follow a two-step procedure. First, a whitened matched filter (WMF) operates on the received signal to leave an effective channel with a monic, causal transfer function. Then, equalization algorithms are used to combat the ISI of the effective channel. Analogously, to combat distortion caused by the MIMO channel (164), MIMO receivers adopt a two-step procedure, namely *spatial* whitened matched filtering followed by MIMO detection to combat the distortion of the effective channel.

11.1.1 Spatial Whitened Matched Filtering

The spatial whitened matched filter (SWMF) aims to make the channel-induced interference spatially causal. However, to achieve causality, the receiver first has to decide the spatial order of the input symbols. The seemingly natural choice is that $\tilde{x}_1$ is the first input symbol, followed by $\tilde{x}_2$, and so on till $\tilde{x}_M$. However, the receiver can choose any other permutation of these symbols as the spatial order of the input symbols. Choosing a spatial order amounts to choosing a $M \times M$ *permutation matrix* $\mathbf{\Pi}$ obtained by permuting the rows of the $M \times M$ identity matrix $\mathbf{I}_M$, such that $\mathbf{x} = \mathbf{\Pi}\tilde{\mathbf{x}}$ is the spatially ordered input

vector. Noting that $\mathbf{\Pi}\mathbf{\Pi}^{\mathrm{T}} = \mathbf{I}_M$ for all permutation matrices, we see that $\tilde{\mathbf{x}} = \mathbf{\Pi}^{\mathrm{T}}\mathbf{x}$, hence $\tilde{\mathbf{H}}\,\tilde{\mathbf{x}} = \tilde{\mathbf{H}}\,\mathbf{\Pi}^{\mathrm{T}}\mathbf{x}$. Substituting this relation in (164), the MIMO channel model takes the form

$$\tilde{\mathbf{y}} = \mathbf{H}'\mathbf{x} + \tilde{\mathbf{n}}, \tag{165}$$

where $\mathbf{H}' = \tilde{\mathbf{H}}\,\mathbf{\Pi}^{\mathrm{T}}$ is the matrix obtained by permuting the columns of $\mathbf{H}$.

With the chosen spatial order of input symbols, the SWMF achieves spatial causality using the Q-L decomposition of $\mathbf{H}'$. More precisely, it obtains a $P \times M$ matrix $\mathbf{Q}$ with orthonormal columns, and a $M \times M$ lower-triangular matrix $\mathbf{L}$ with positive real diagonal elements, such that $\mathbf{H}' = \mathbf{QL}$. The Q-L decomposition can be performed only if there are more channel outputs than inputs, i.e., $P \geq M$. Through the remainder of this chapter, we assume this is the case. Under this assumption, the Q-L decomposition can be performed using Gram Schmidt orthonormalization of the columns of $\mathbf{H}'$.

After Q-L decomposition, the SWMF multiplies the received vector $\tilde{\mathbf{y}}$ by $\mathbf{Q}^*$, the conjugate transpose of $\mathbf{Q}$. Substituting $\mathbf{Q}^*\mathbf{H}' = \mathbf{Q}^*\mathbf{QL} = \mathbf{L}$ in (165), the SWMF output is

$$\mathbf{y} = \mathbf{Q}^*\tilde{\mathbf{y}} = \mathbf{Lx} + \mathbf{n}. \tag{166}$$

It is easy to show that the effective noise vector $\mathbf{n} = \mathbf{Q}^*\tilde{\mathbf{n}}$ has independent, complex Gaussian entries of variance N_0. Note that the effective channel (166) is indeed spatially causal, i.e., the i^{th} element of $\mathbf{y}$, namely

$$y_i = l_{ii}x_i + \sum_{j=1}^{i-1} l_{ij}x_j + n_i, \tag{167}$$

is a noisy scaled version of the current symbol x_i, with interference only from *past* symbols $x_1, x_2, \ldots, x_{i-1}$.

The spatial causality of the effective channel greatly facilitates detection, namely the task of estimating the ordered channel input vector $\mathbf{x}$ from the effective channel output $\mathbf{y}$. In this chapter, we will discuss a variety of detection algorithms, which exploit the spatial causality of the effective channel. Before developing detection algorithms, we note that their performance clearly depends on the effective channel's transfer matrix $\mathbf{L}$. Now, the choice of the spatial order determines $\mathbf{H}' = \tilde{\mathbf{H}}\boldsymbol{\Pi}^{\mathrm{T}}$, and hence the transfer matrix $\mathbf{L}$. Consequently, the spatial order also determines the performance of detection algorithms.

We will discuss the performance of detection algorithms for two choices of the spatial order. The first choice is the natural order, namely the permutation matrix $\boldsymbol{\Pi}$ is the identity matrix $\mathbf{I}_M$, irrespective of the value of the MIMO channel matrix $\tilde{\mathbf{H}}$. The second ordering choice is the one used in the popular V-BLAST receiver [11], summarized below.

> **Remark 7.** Given $\tilde{\mathbf{H}}$, the V-BLAST receiver chooses a permutation matrix $\boldsymbol{\Pi}_{\mathrm{VB}}(\tilde{\mathbf{H}})$ that maximizes the minimum value among $l_{11}, l_{22}, \ldots, l_{MM}$. This choice is made in a greedy, sequential fashion as follows [11]. Of the M possible choices of the first input (equivalently, the first row of $\boldsymbol{\Pi}_{\mathrm{VB}}(\tilde{\mathbf{H}})$), the V-BLAST receiver chooses the input that maximizes the value of l_{11}. Given this choice, the second input is chosen to maximize l_{22}, and so on.

Obtaining the V-BLAST spatial order requires more computation than merely using the natural order. In return, we will see that V-BLAST ordering increases the accuracy and/or reduces the complexity of many detection algorithms. Computationally efficient methods to obtain the V-BLAST spatial order are given in [61].

11.1.2 Successive Cancellation and ML Detectors

Different detection algorithms, also known as detectors, can be used to obtain an estimate $\hat{\mathbf{x}}$ of the ordered input vector $\mathbf{x}$, given $\mathbf{y}$ and $\mathbf{L}$. The accuracy of a detector is measured by its *word error rate* (WER), namely the probability that $\hat{\mathbf{x}} \neq \mathbf{x}$. The twin goals of detector design are to achieve low WER and to maintain a low computational complexity. However, there is a trade-off between these twin goals, as illustrated by the two detectors discussed in this section, namely the *successive cancellation* (SC) and *maximum likelihood* (ML) detectors.

The SC detector is analogous to the decision feedback equalizer used to combat ISI in SISO channels [23]. The SC detector performs detection in M stages. In the i^{th} stage, x_i is estimated from y_i, after cancelling off the estimated interference from past symbols. More precisely, the SC detector obtains the decision metric

$$y_i' = y_i - \sum_{j=1}^{i-1} l_{ij} \hat{x}_j^{\text{SC}} \tag{168}$$

using estimates $\{\hat{x}_j^{\text{SC}}\}$ from past stages. If these past estimates are accurate, then (167) implies $y_i' = l_{ii} x_i + n_i$. Assuming this is the case, the SC detector obtains $\hat{x}_i^{\text{SC}}$ by *slicing* y_i'/l_{ii}, i.e., rounding it off to the nearest symbol in the symbol alphabet $\mathcal{A}$. Note that an SC detector is just a cascade of M slicers, and is therefore computationally simple. However, it suffers from high WER, i.e., with high probability, $\hat{\mathbf{x}}^{\text{SC}} \neq \mathbf{x}$. Intuitively, the reason for the high WER of the SC detector is that it estimates the symbol x_i using y_i alone, without using the information contained about x_i in future symbols $y_{i+1}, \ldots, y_M$.

In contrast to the SC detector, the ML detector optimally uses the available information, and achieves the minimum WER among all possible detectors [7]. The ML detector uses the conditional probability density function of $\mathbf{y}$ given that the unknown channel-input vector is some $\mathbf{z} \in \mathcal{X}$, namely

$$p(\mathbf{y} \mid \mathbf{z}) = \frac{1}{(\pi N_0)^T} \exp\left(\frac{-\|\mathbf{y} - \mathbf{L}\mathbf{z}\|^2}{N_0} \right). \tag{169}$$

The ML detector's estimate $\hat{\mathbf{x}}^{\mathrm{ML}}$ is the vector in the transmit alphabet $\mathcal{X}$ with the maximum value of $p(\mathbf{y} \mid \mathbf{z})$. Note that $p(\mathbf{y} \mid \mathbf{z})$ is a decreasing function of

$$J(\mathbf{z}) = \|\mathbf{y} - \mathbf{L}\mathbf{z}\|^2, \tag{170}$$

which is called the ML *cost function* of $\mathbf{z}$. Consequently, the ML detector's estimate is the least-cost vector in the input alphabet $\mathcal{X}$, namely

$$\hat{\mathbf{x}}^{\mathrm{ML}} = \underset{\mathbf{z} \in \mathcal{X}}{\operatorname{argmin}} \; J(\mathbf{z}). \tag{171}$$

While the ML detector has low WER, it incurs a heavy computational burden. For instance, one way to implement ML detection is to compute the costs $J(\mathbf{z})$ for all $\mathbf{z} \in \mathcal{X}$, and pick out the cheapest vector. However, there are often thousands of vectors in $\mathcal{X}$, and such enumeration is impractical. Instead, we aim to develop less computation-demanding algorithms to implement MIMO detection. To achieve this goal, it will prove useful to represent the detection problem graphically, using the *detection tree*.

11.2 ML Detection Is a Tree-Search Problem

In this section, we show that ML detection amounts to the search for the cheapest leaf node in the detection tree. We begin by describing the detection tree itself. The tree starts with the left-most node, or *root*. It consists of M stages, one for each input symbol. The root node is connected to $|\mathcal{A}|$ child nodes, one for each value of the first symbol z_1. Each of these nodes is connected to $|\mathcal{A}|$ child nodes depending on z_2, and so on. Thus, for each possible channel-input vector $\mathbf{z} \in \mathcal{X}$, there is a unique path through the tree that begins at the root and ends at one of the right-most nodes, known as *leaf* nodes. As an illustrative example, the detection tree for the case of $M = 2$ inputs with an input alphabet of $\mathcal{A} = \{\pm 1\}$ is shown in Fig. 15. The bold-faced label below each node of depth one shows the choice of z_1. The label for the leaf nodes shows the corresponding choice (z_1, z_2).

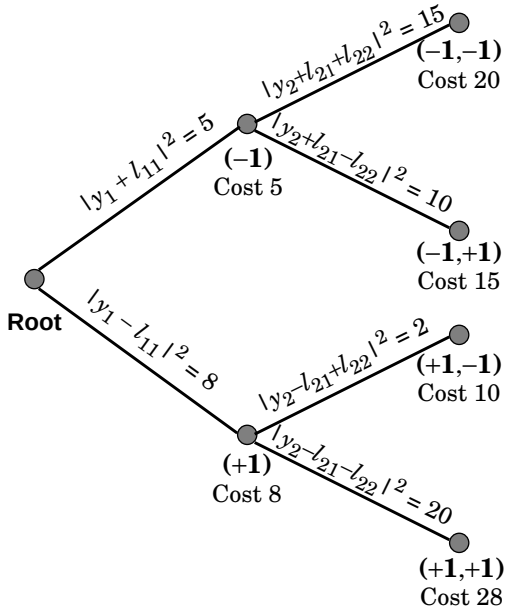

Fig. 15. Illustration of the detection tree for a channel with $M = 2$ inputs and $\mathcal{A} = \{\pm 1\}$.

In terms of the detection tree, the ML cost function $J(\mathbf{z})$ can be interpreted as the sum of N branch costs, one for each branch on the path corresponding to $\mathbf{z}$. More precisely, consider the branch in the i^{th} stage, connecting the nodes $(z_1, z_2, \ldots, z_{i-1})$ and $(z_1, z_2, \ldots, z_i)$. If its cost is defined as

$$B(z_1, z_2, \ldots, z_i) = \left| y_i - \sum_{j=1}^{i-1} l_{ij} z_j \right|^2, \tag{172}$$

it is easy to see from (170) that $J(\mathbf{z}) = \| \mathbf{y} - \mathbf{L}\mathbf{z} \|^2$ is just the sum $B(z_1) + B(z_1, z_2) + \ldots + B(z_1, z_2, \ldots, z_N)$.

An intuitive interpretation of the branch cost is obtained by splitting (172) as $|y_i' - l_{ii} z_i|^2$, where $y_i' = y_i - \sum_{j=1}^{i-1} l_{ij} z_j$ is the cancellation residue used by the SC detector. In other words, $B(z_1, z_2, \ldots, z_i)$ represents the cost of choosing the symbol z_i, after cancelling out interference in y_i from the already chosen symbols $z_1, z_2, \ldots, z_{i-1}$. Extending this interpretation, the cost of a node is defined as the cost of all the choices it represents, i.e., it is the sum of the costs of all the branches connecting that node to the root node, namely

$$C(z_1, z_2, \ldots, z_i) = \sum_{j=1}^{i} B(z_1, z_2, \ldots, z_i). \tag{173}$$

Equivalently, node costs can be defined recursively as

$$C(z_1, z_2, \ldots, z_i) = C(z_1, z_2, \ldots, z_{i-1}) + B(z_1, z_2, \ldots, z_i), \tag{174}$$

with the root node defined to have cost zero. Clearly, the cost of the leaf node $(z_1, z_2, \ldots, z_M)$ is equal to the ML cost function $J(\mathbf{z})$. For illustration, each branch in Fig. 15 is labeled with its branch cost formula (172) after substituting the values of z_1 and z_2 corresponding to the branch. For illustration, we have arbitrarily assigned some values to the branch costs. The corresponding node costs are shown below the node labels.

Having defined branch and node costs on the detection tree, we now revisit the detectors described in the previous section. Each detector outputs an estimate from the transmit alphabet $\mathcal{X}$, or equivalently a leaf node on the tree.

The SC detector starts from the root node and always moves forward on the cheapest available branch, till a leaf node is reached. The leaf node so obtained is the SC detector's output. In the example of Fig. 15, the SC detector first takes the cheapest branch to reach the node (-1), and then again takes the cheapest branch forward to reach the leaf node $(-1, +1)$, whose cost is 20.

Since the cost of each leaf node is equal to the ML cost function, the ML detector's estimate is the cheapest leaf node in the detection tree. For instance, in the tree of Fig. 15, the ML detector finds the cheapest leaf node, namely $(+1, -1)$, whose cost is 10. Note that the greedy SC detector, which stitches together locally cheap branches, does not necessarily reach the cheapest leaf node in the tree. To implement ML detection, we need a tree-search algorithm that seeks out the cheapest node in the tree. In the next section, we introduce the basic operations of tree-search algorithms, and also introduce a measure of the computational complexity of such algorithms. We also present a simple search strategy that enables the design of computationally efficient tree-search algorithms to implement ML detection.

11.3 Introduction to Tree-Pruning Algorithms

Tree-search algorithms store and manipulate the nodes of the detection tree. All tree-search algorithms start at the root node of the tree. Subsequently, they access or *visit* other nodes in the tree, by making *forward moves* on the tree's branches. More precisely, tree-

search algorithms treat a node as a data structure N, with five fields. The first two fields of a node are label fields, namely the node's depth i and its branch index history $(z_1, z_2, \ldots, z_i)$. The third field is the node's cost $C(z_1, z_2, \ldots, z_i)$. Fourth, the cancellation residue

$$y_{i+1}' = y_{i+1} - \sum_{j=1}^{i} l_{i+1,j} z_j \tag{175}$$

is stored as a field, to facilitate computation of the costs of successor branches of N. Finally, the cost B_{i+1} of the branch on which the last forward move was made from the node N is also stored as a field. If no forward move has yet been made from N, B_{i+1} is initialized to ∞. For example, the fields of the root node at the start of processing are: depth 0, no branch index history, cost 0.0, cancellation residue $y_1' = y_1$, and $B_1 = \infty$.

If a tree-search algorithm has already visited node N and wishes to visit a child node M of N, it only needs to compute and store the fields of the node M. Suppose the branch connecting N to M has label z_{i+1}. Then, the depth and branch index history of M are clearly $i + 1$ and $(z_1, z_2, \ldots, z_{i+1})$ respectively. The cancellation residue y_{i+1}' of N is used to compute the branch cost $B(z_1, z_2, \ldots, z_{i+1}) = |y_{i+1}' - l_{i+1,i+1} z_{i+1}|^2$. Then, the cost of M is the sum of the branch cost $B(z_1, z_2, \ldots, z_{i+1})$ and the cost of N. The cancellation residue y_{i+2}' of M is obtained by the formula (175). Since no successor of M has been visited, the last-forward-branch cost B_{i+2} of M is set to ∞. Finally, the last-forward-branch cost B_{i+1} of N is updated to $B(z_1, z_2, \ldots, z_{i+1})$, to reflect the latest forward move from N.

We use the number of nodes visited by an algorithm as a measure of its computational complexity. This is a more tractable measure than implementation-dependent quantities like the number of flops required, or the processing time. Now, any detector has to output a valid vector in the alphabet $\mathcal{X}$, hence the corresponding tree-search algorithm has to

output a leaf node. Now, in order to reach a leaf node starting from the root node, the algorithm has to visit at least M nodes. The SC detection algorithm visits exactly M nodes, and is therefore the tree-search algorithm of least complexity. However, as we have already seen, the SC algorithm does not necessarily find the cheapest leaf node in the tree. One way to find the cheapest leaf node is to run an exhaustive search algorithm, which visits all the nodes in the tree. However, this is computationally daunting. For example, if the MIMO channel has $M = 8$ inputs symbol alphabet $\mathcal{A}$ is a 16-QAM alphabet, there are a total of $(1 + |\mathcal{A}| + \ldots + |\mathcal{A}|^M) = 4{,}581{,}298{,}449$ nodes in the tree. Visiting all these nodes is clearly impractical.

In this chapter, we discuss *tree-pruning* (TP) algorithms, which find the cheapest leaf node without visiting all the nodes in tree. There are a variety of TP algorithms, but all of them use the same strategy to avoid exhaustive search. In the remainder of this section, we describe this basic strategy.

Consider a genie-aided tree-search algorithm, which knows only the cost $C_{\min}$ of the cheapest leaf node $\hat{\mathbf{x}}^{\mathrm{ML}}$, but not its label. To implement ML detection, this genie-aided algorithm must explore the tree and find out $\hat{\mathbf{x}}^{\mathrm{ML}}$. Now, suppose this algorithm is at some node N, and is seeking to move forward so as to visit new nodes. Note that branch costs are non-negative, hence node costs are non-decreasing as one moves deeper into the tree. Consequently, if a child M of N has cost greater than $C_{\min}$, then all its descendants also have cost greater than $C_{\min}$. In particular, none of these descendants can be the cheapest leaf node $\hat{\mathbf{x}}^{\mathrm{ML}}$. Since the tree-search algorithm only aims to find $\hat{\mathbf{x}}^{\mathrm{ML}}$, can avoid visiting M and all its descendants. Note that the cost of M is less than $C_{\min}$ if and only if the cost of the branch connecting M to N has cost less than the upper bound $U_{\mathsf{N}} = C_{\min} - C_{\mathsf{N}}$, where

C_{N} is the cost of the node N. Thus, at any node N, if the tree-search algorithm moves forward only on branches with cost lesser than U_{N}, it automatically ensures that nodes with cost greater than $C_{\min}$ are never visited. Equivalently, the algorithm *prunes* out branches with cost greater than U_{N} before a forward move from node N.

Like the above genie-aided algorithm, tree-pruning algorithms also implement pruning of branches before any forward move. However, unlike the genie-aided algorithm, they do not know the actual cost $C_{\min}$ of the cheapest leaf node. Instead, TP algorithms maintain a *threshold* T, which is an estimate of the cheapest leaf node's cost. At every node, branches with cost greater than $T - C_{\mathsf{N}}$ are pruned before any forward move. The threshold T is initialized as a finite quantity C_0. Subsequently, the TP algorithm searches through the tree, visiting only nodes with cost less than C_0. Note that if $C_0 < C_{\min}$, the TP algorithm will finish searching through the tree without visiting any leaf node. When this happens, the TP algorithm senses an *erasure* and restarts the search with a higher value of the initial threshold C_0. Eventually, C_0 is larger than $C_{\min}$, and the TP algorithm visits some leaf node, say L. Then, the TP algorithm estimates that L is the cheapest leaf node, and *tightens* the threshold to the cost of L. The tightening of the threshold helps the TP algorithm to prune out more nodes later in the search process. As the process continues, the TP algorithm visits cheaper and cheaper leaf nodes. At the end of the search, the cheapest leaf node $\hat{\mathbf{x}}^{\mathrm{ML}}$ has been found.

The choice of the initial threshold C_0 is crucial. Ideally, we would like to avoid erasure by choosing $C_0 > C_{\min}$. On the other hand, the lower the value of C_0, the fewer the number of nodes visited. Thus, in order to both avoid erasures and minimize node visits, C_0 should be a good estimate of the actual value of $C_{\min}$. To obtain this estimate, note that with high

probability, $\hat{\mathbf{x}}^{\mathrm{ML}}$ is the actual transmit vector $\mathbf{x}$, whose cost is the noise energy $\|\mathbf{n}\|^2$. Now, though the noise energy is unknown, its average value, namely MN_0, is known. Thus, one good initial threshold is $C_0 = \alpha MN_0$. The correction factor α accounts for the fact that the noise energy will sometimes exceed its mean value. In spite of the correction factor, erasures do occur at times. In this case, we suggest scaling the initial threshold C_0 by a factor β before restarting the search. The values of α and β determine the number of nodes visited by a TP algorithm, and must be chosen carefully. Heuristically, we have found that $\alpha = 2.0$ works well, when the number of channel inputs M is less than or equal to 4. When $M > 4$, we suggest $\alpha = 1.5$. For both cases, we suggest $\beta = 1.5$. The actual computation-minimizing values of α and β depend on the channel model, SNR, the symbol alphabet and the value of N. Given all these parameters, the optimum values can be obtained by trial-and-error.

To sum up, the basic computation-reduction strategy of TP algorithms has two components, namely *threshold maintenance* and *branch pruning*.

- The threshold is an estimate of the cheapest leaf node's cost. Threshold maintenance involves two tasks. At the start of the search, the threshold is *initialized* based on an statistical estimate of the cheapest leaf node cost. Second, whenever a leaf node is reached, the threshold is *tightened* to the cost of the leaf node.

- Branch pruning is performed before every forward move, and ensures that nodes with cost greater than threshold are not visited.

Using this two-fold strategy, TP algorithms efficiently find the cheapest leaf node, provided the initial threshold is higher than the actual cost of cheapest leaf node. If not, the

TP algorithm senses an erasure, and iteratively increases the initial threshold till it is greater than the cheapest leaf node's cost.

The above basic strategy only defines the operations to be done before every forward move, and when a leaf node is reached. It can be implemented in multiple ways. To see this more clearly, note that if the TP algorithm has already visited multiple nodes, it needs to choose one of these nodes as the site of the next forward move. Depending on how this choice is made, we get multiple TP algorithms. For example, *depth-first* TP algorithms always choose the deepest of all available nodes to attempt the next forward move. In contrast, *breadth-first* TP algorithms choose one of the shallowest available nodes as the site for the next move. In the next two sections, we develop precise depth-first and breadth-first TP algorithms.

11.4 Depth-First Tree-Pruning: The Sphere Decoder

The depth-first tree-pruning algorithm that we develop in this section is already well known in the MIMO detection literature, by the name of the *sphere decoder* [25][26][27]. The sphere decoder presented in the literature often assumes that the symbol alphabet $\mathcal{A}$ is a *lattice* alphabet containing real integers. Instead, we present a more general version, where $\mathcal{A}$ is some finite, complex alphabet.

The operation of the sphere decoder proceeds in processing cycles. In each processing cycle, the focus of operation is the deepest available node, called the *current node*. At the start of the search, the current node is the root node. In each processing cycle, the sphere decoder checks to see if the current node has any unpruned child branches. If so, it moves forward on the cheapest branch to a child node, and the next processing cycle begins. If

there are no unpruned child branches (this could be because the current node is a leaf node with no child branches, in which case the threshold is first tightened), the sphere decoder just moves back to the parent of the current node. In the next cycle, it looks for an unpruned, *unexplored* child branch of the parent node. If there are such branches, it moves forward on the cheapest one. Otherwise, it moves back again. Proceeding thus, the sphere decoder explores the entire tree. When it finds that the root node has no remaining unexplored branches, it recognizes that the search is complete. If no leaf node has been visited, the sphere decoder detects erasure and repeats the search with a higher initial threshold. Otherwise, it outputs the cheapest leaf node visited, and quits.

In Fig. 16, we present a precise pseudocode for the sphere decoder. The variable i represents the depth of the current node, i.e., the node from which a forward move is being attempted in that processing cycle. In addition to the current node N_i, the sphere decoder also stores the nodes N_1, N_2, …, N_{i-1} of depths 1, 2, …, i respectively, which lie on the path connecting N_i to the root node. These are needed because the sphere decoder moves to N_{i-1} after exploring all the descendants of N_i, then to N_{i-2} after exploring all the descendants of N_{i-1}, and so on. For convenience, the root node is denoted N_0. In each processing cycle, the threshold is updated if N_i is a root node. Then, the sphere decoder executes a *seek* step, looking for unexplored, unpruned child branches of N_i. If any are found, it moves forward on the cheapest one, and visits N_{i+1} for the next cycle. If no branches are found, i is decrements and N_{i-1} is the current node for the next processing cycle. The other steps are self-explanatory, when read with the bold-faced comments preceding them.

Input — $M \times 1$ complex vector $\mathbf{y}$, $M \times M$ lower triangular matrix $\mathbf{L}$ with non-negative, real diagonal entries, finite symbol alphabet $\mathcal{A}$
Output — $M \times 1$ complex vector $\hat{\mathbf{x}}$ with elements in $\mathcal{A}$ such that $\|\mathbf{y} - \mathbf{L}\hat{\mathbf{x}}\|2$ is minimum

Start Up $C_0 = \alpha N N_0$, SomeLeafNodeVisitedFlag = OFF
Initialize Root Node
Root.depth = 0, Root.cost = 0.0, Root.y' = $y1$, Root.lastforwardcost = ∞
Initialize Search $T = C_0$
N_0.depth = Root, $i = 0$ /* Deepest available node is root node, depth 0 */
Process Current Node
If current node is leaf node, tighten threshold and update ML decision
if $i == N$
SomeLeafNodeVisitedFlag = ON /* Some leaf node visited, so no erasure */
$T = N_i$.cost
$\hat{\mathbf{x}} = N_i.[z_1, z_2, \ldots, z_N]^T$
$i = i - 1$ /* Leaf node processed, just move back */
endif
Seek Cheapest Unexplored, Unpruned Branch
LowerBound = N_i.lastforwardcost /* Do not reconsider explored branches */
UppperBound = $T - N_i$.cost
Look for z^* such that $|N_i.y' - l_{i+1,\, i+1}z|^2$ is minimum among all z in $\mathcal{A}$ such that
LowerBound $< |N_i.y' - l_{i+1,\, i+1}z|^2 \leq$ UpperBound

If possible, move forward on cheapest branch
if any z^* found in last step
Move forward on branch labeled z^* and visit child N_{i+1}
$i = i + 1$ /*Increment highest depth after forward move */
goto **Process Current Node**
endif
If forward move not possible, move back if possible
if $i > 1$
$i = i - 1$ /* Move back by reducing depth*/
goto **Process Current Node**
endif
If backward move is also not possible, search is over. Repeat search or quit.
if SomeLeafNodeVisitedFlag = OFF
$C_0 = \beta C_0$ /* Erasure has occurred, expand threshold and repeat*/
goto **Initialize Search**
else output $\hat{\mathbf{x}}$ and quit

Fig. 16. Sphere Decoder: depth-first tree-pruning algorithm for efficient ML detection.

We close the discussion of the sphere decoder with some remarks about the *seek* step in Fig. 16, which searches for the cheapest unpruned, unexplored branch for the next forward move. By placing a lower bound equal to the cost of the last forward move, the search is restricted only to those successor branches which have not already been

explored. Further, since the cheapest unexplored branch is picked, the algorithm always explores successor branches of any node in increasing order of cost. In principle, one can explore successor branches in any order. In fact, the original sphere decoder [24] explored branches in increasing order of their index value. However, exploring successor branches in increasing order of cost enables the sphere decoder to visit cheap leaf nodes early, and hence tighten the threshold. This leads to more effective pruning subsequently, and reduces the overall number of nodes visited by the algorithm. Cost-based ordering of branches was first suggested in [29], and has been almost universally adopted since.

For some symbol alphabets, there are simple ways to pick the cheapest unexplored branch. For example, suppose the symbols are drawn from an 8-PAM alphabet, consisting of all odd integers from –7 to +7. In the first move from node N_i, one can obtain the cheapest branch index by slicing $y_{\text{eff}} = \mathsf{N}_i.y' / l_{i+1,\, i+1}$, or equivalent rounding it off to the nearest odd integer between –7 to +7. In every subsequent forward move, the symbol corresponding to the cheapest unexplored branch alternates around the sliced value, unless it exceeds the maximum limit of ±7. For instance, $y_{\text{eff}} = 3.2$, the cheapest branch index in the first visit is $z^* = 3$. In subsequent visits, the cheapest branch index alternates around the central value of 3, taking on the values 5, 1, 7, –1, –3, –5, and finally –7. This example can extended to a general PAM alphabet containing all odd integers in the range $[-z_{\max}, z_{\max}]$, as discussed in [30]. When the symbols are drawn from a complex QAM alphabet, [62] suggests a look-up based ordering implementation. Such alphabet-specific implementations are useful in reducing the complexity of the seek step, but they are not essential. The efficiency of sphere decoding is primarily because of branch-pruning, and threshold management.

11.5 Modified Breadth-First Tree-Pruning: The Hybrid Decoder

In the last section, we discussed the sphere decoder, which is a depth-first tree-pruning algorithm. In this section, we will develop a new tree-pruning algorithm, called the *hybrid decoder*, which combines the features of breadth-first and depth-first tree search strategies. A slight modification to the hybrid decoder yields the *bounded stack decoder*, which allows a flexible trade-off between the two conflicting goals of MIMO detector design, low error rate and low computational complexity.

We first develop a purely breadth-first TP algorithm, and point out that it is inherently defective. Recall that a breadth-first TP algorithm always picks the shallowest available node as the site of the next forward move. In effect, it starts with the root node, and prunes out all branches with cost less than the initial threshold C_0. Then, it successively makes forward moves on all the unpruned successor branches, till it assembles a *stack* $\mathcal{S}_1$, containing all nodes of depth one with cost less than C_0. Then, it visits the unpruned successors of nodes in $\mathcal{S}_1$, till eventually it obtains a stack $\mathcal{S}_2$ of nodes in the second level. Thus, the breadth-first algorithm just generates stacks $\mathcal{S}_i$ containing all nodes in depth i with cost less than the threshold C_0. The stacks are generated recursively, i.e., $\mathcal{S}_{i+1}$ is obtained by just visiting all the unpruned successors of all nodes in $\mathcal{S}_i$. Proceeding thus, the algorithm eventually starts exploring the successors of nodes in $\mathcal{S}_{N-1}$, which are leaf nodes (if there are any leaf nodes with cost less than C_0). Every time a leaf node is visited, it tightens the threshold. Eventually, all the successors of nodes in $\mathcal{S}_{M-1}$ are explored, and the algorithm finds the cheapest leaf node. Note that the threshold is tightened only towards the end of all processing, while assembling $\mathcal{S}_M$. While assembling all the earlier stacks, the

threshold remains at its initial value C_0. Consequently, the breadth-first algorithm visits all non-leaf nodes whose cost is less than C_0. In contrast, the sphere decoder visits leaf nodes early, tightens the threshold, and visits fewer nodes subsequently.

We propose a modification to the breadth-first algorithm, to make it tighten the threshold by seeking out cheap leaf nodes early. The modified algorithm, which we call the *hybrid* decoder, is neither breadth-first nor depth-first, but a hybrid of the two. Similar to the breadth-first algorithm, the hybrid decoder also starts with the stack $\mathcal{S}_0$ containing only the root node, and recursively generates stacks $\mathcal{S}_i$ for future levels. However, every time it visits a new node, it not only adds the new node to stack $\mathcal{S}_i$, but also performs successive cancellation starting from the new node. More precisely, suppose the hybrid decoder takes up a node N_{i-1} from stack $\mathcal{S}_{i-1}$, and adds its unpruned successor N_i to stack $\mathcal{S}i$. After this, the breadth-first TP algorithm would proceed to visit the next unpruned successor of N_{i-1}. Instead, the hybrid decoder starts from N_i and successively moves forward on the cheapest unpruned successor branch of the current node. Thus, it visits nodes N_{i+1}, N_{i+2}, …, and adds them to stacks $\mathcal{S}_{i+1}$, $\mathcal{S}_{i+2}$, … respectively. This forward movement stops when either all successor branches of some N_j get pruned out, or a leaf node is reached. In the latter case, the algorithm tightens the threshold. After forward movement stops, the hybrid search algorithm returns to node N_{i-1} from stack $\mathcal{S}_{i-1}$, and seeks its next unpruned successor. A precise pseudocode of the hybrid search algorithm is shown below, in Fig. 17.

Input — $M \times 1$ complex vector $\mathbf{y}$, $M \times M$ lower triangular matrix $\mathbf{L}$ with non-negative, real diagonal entries, finite symbol alphabet $\mathcal{A}$
Output — $M \times 1$ complex vector $\hat{\mathbf{x}}$ with elements in $\mathcal{A}$ such that $\|\mathbf{y} - \mathbf{L}\hat{\mathbf{x}}\|^2$ is minimum

Start Up $C_0 = \alpha MN_0$, SomeLeafNodeVisitedFlag = OFF
Initialize Root Node
Root.depth = 0, Root.cost = 0.0, Root.y' = $y1$, Root.lastforwardcost = ∞
Initialize Search $T = C_0$, $\mathcal{S}_0$ = {Root}
for i = 1: N
Generate stack i from current stack i–1
while ($\mathcal{S}_{i-1}$ not empty)
N_{i-1} = Top node in $\mathcal{S}_{i-1}$
Seek Next BranchOf Top Node
Lower Bound = N_{i-1}.lastforwardcost /* Remove explored branches */
Upper Bound = $T - \mathsf{N}_{i-1}$.cost /* Branch Pruning*/
Look for z_i such that $|\mathsf{N}_{i-1}.y' - l_{ii}z_i|^2$ is minimum among all z in $\mathcal{A}$
such that Lower Bound < $|\mathsf{N}_{i-1}.y' - l_{ii}z|^2 \leq$ Upper Bound
if no branch found, goto **Top Node Fully Explored**
Visit new node, start successive cancellation
Move forward from N_{i-1} on branch labeled z_i to visit child N_i
$k = i$
Push Next Node Push N_k to stack $\mathcal{S}_k$
if ($k < N$)
Seek cheapest branch forward
UpperBound = (T – N_k. cost)
z_{k+1} = Index of cheapest successor branchof N_k
if $|\mathsf{N}_k.y' - l_{k+1,\,k+1}z_{k+1}|^2$ < Upper Bound
Move forward from N_k on branch labeled z_{k+1} to visit N_{k+1}
Increment k
goto **Push Next Node**
endif
else goto **Successive Cancellation Done**
endif
else
Leaf Node Reached, Tighten Threshold
SomeLeafNodeVisitedFlag = ON
$T = \mathsf{N}_k$.cost
$\hat{\mathbf{x}} = \mathsf{N}_k.[z_1, z_2, \ldots, z_N]^{\mathrm{T}}$
goto **Successive Cancellation Done**
endelse
Successive Cancellation Done
goto **Seek Next Branch Of Top Node**
Top Node Fully Explored
Remove top node from $\mathcal{S}_{i-1}$
Loops till $\mathcal{S}_{i-1}$ Empty endwhile
Loops till $i = N$ endfor
Search Over. Repeat Search Or Quit
if (SomeLeafNodeVisitedFlag = OFF)
$C_0 = \beta C_0$ /* Erasure has occurred, expand threshold*/
goto **Initialize Search**
endif
else Output $\hat{\mathbf{x}}$ and Quit

Fig. 17. Hybrid search algorithm for efficient ML detection.

The hybrid and sphere decoders have different movement patterns on the detection tree. Therefore, the number of nodes visited by the two algorithms during the detection process are different. In Section 11.7.3, we will see that the two algorithms visit roughly the same number of nodes visited, though there are some small differences. Now, we discuss the more significant differences between the two algorithms, namely their relative suitability for high-speed parallel implementation, and their memory requirements.

The hybrid decoder is more suited to parallel implementation than the sphere decoder. In each processing cycle, the sphere decoder moves one stage forward or backward from the current node. Since all the processing is focused on just one node, implementation by multiple processors needs some complicated management mechanism. In contrast, the basic unit of updation for the hybrid decoder is not the node, but the stack. Now, while assembling stack $\mathcal{S}_i$, different nodes from stack $\mathcal{S}_{i-1}$ are extended, i.e., their unpruned successors are visited, and successive cancellation is done on each of them. The extension of different nodes in $\mathcal{S}_{i-1}$ can be done simultaneously and independently by multiple processors. For a thorough discussion of parallel implementations of tree-search algorithms, we refer the reader to [63].

Comparing the memory requirement of the two algorithms, the sphere decoder has a distinct advantage over the hybrid decoder. Recall that the sphere decoder only needs to store the nodes N_0, N_1, …, N_i when it is currently at depth i. Since the largest depth is $i = M$, the sphere decoder only needs $N + 1$ nodes of memory to cover the worst case. In contrast, the hybrid decoder of Fig. 17 needs memory allocated for the stacks $\mathcal{S}_0$, $\mathcal{S}_1$, …, $\mathcal{S}_M$. Memory allocation for the hybrid decoder is problematic for two reasons. Firstly, the actual size of each stack $\mathcal{S}_i$ is random, because it depend on the number of nodes of depth i

with cost less than threshold, and node costs are random. Secondly, there are $|\mathcal{A}|^i$ nodes in level i. Even if a small fraction of these nodes have cost less than threshold, there could be a few thousand nodes in $\mathcal{S}_i$. Thus, the memory requirement of the stack decoder is both unpredictable and large.

11.5.1 The Bounded Stack Hybrid Decoder

One way to sidestep the unpredictable and large memory requirement of the hybrid decoder is to simply restrict the maximum size of each stack to some fixed pre-determined value, say $S_{\max}$ nodes. This restriction yields what we call the *bounded stack* hybrid decoder, and can be implemented by a simple change to the hybrid decoder algorithm of Fig. 17. Before pushing a new node into a stack $\mathcal{S}_k$, we check the size of $\mathcal{S}_k$. If $\mathcal{S}_k$ has fewer than $S_{\max}$ nodes, the new node is added to $\mathcal{S}_k$. Otherwise, the cost of the new node is compared to the cost of the costliest node in $\mathcal{S}_k$. If the new node has lower cost, it replaces the costliest node in $\mathcal{S}_k$. On the other hand, if the new node has higher cost, it is discarded, without being stored anywhere. Thus, the stack $\mathcal{S}_k$ now contains only the cheapest $S_{\max}$ nodes of depth k with cost less than the threshold. If there are more than $S_{\max}$ sub-threshold nodes, the other nodes are discarded.

Note that the descendants of the discarded nodes are not visited, even though their cost is less than threshold. If one of these unvisited descendants is the cheapest leaf node, the hybrid stack decoder will not find it. Thus, the bounded stack hybrid decoder does not necessarily find the cheapest leaf node, even if there is no erasure. If the stack limit $S_{\max}$ is small, the memory required by the decoder is low. On the other hand, a small value of $S_{\max}$ implies more sub-threshold nodes are discarded, hence increases the deviation from

ML detection. In particular, if $S_{\max} = 1$, $\mathcal{S}_1$ contains only the cheapest child of the root node, $\mathcal{S}_2$ contains only the cheapest child of $\mathcal{S}_1$, and so on. In other words, with a stack size limit of one, the stack limited hybrid decoder reduces to the successive cancellation detector. Thus, one can think of $S_{\max}$ as a parameter that controls the WER-complexity trade-off: as $S_{\max}$ increases, the WER decreases but the complexity and the memory requirement increase. However, we will see in Section 11.7.1 that even low values of $S_{\max}$ are sufficient to closely approach the error rate of ML detection.

11.6 The Complexity of Tree-Pruning Algorithms

We have presented two tree-pruning algorithms that perform ML detection, namely the hybrid and sphere decoders. In this section, we make some general remarks about the computational complexity of these algorithms. In the next section, simulation results are presented to illustrate these remarks.

As already mentioned, the computational complexity of tree-pruning algorithms is measured by the number of nodes visited. The SC detector always visits exactly M nodes. On the other hand, a tree-pruning algorithm visits all nodes, except those that are eliminated by cost-based pruning at some stage. Note that the node costs are functions of the effective channel output $\mathbf{y}$ and the lower-triangular transfer matrix $\mathbf{L}$, both of which are random. Therefore, the number of nodes visited by a tree-pruning algorithm is also random. One way to quantify the computational complexity is to obtain the distribution of the random number of nodes visited. However, the entire distribution is cumbersome to obtain and analyze. Instead, one can analyze the mean of the distribution, namely the *average* number of nodes visited, which gives the *average* complexity of a TP algorithm.

Clearly, the average number of nodes visited by a TP algorithm depends on the details of the algorithm itself. For example, the sphere and hybrid decoders differ in the number of nodes visited for the same tree. However, for all tree-pruning algorithms, the average number of node visits follows some general trends. For instance, it is a function of the distribution of $\mathbf{L}$ and $\mathbf{y}$; the symbol alphabet $\mathcal{A}$; the MIMO channel dimensions P and M. Further, the distribution of $\mathbf{L}$ depends both on the distribution of the MIMO channel matrix $\mathbf{H}$ and the spatial order used by SWMF (natural or V-BLAST). The distribution of $\mathbf{y}$ depends on the noise variance N_0, or equivalently the SNR S. Considering the number of parameters involved, it is clear that precise analysis of the average number of node visits for a given tree-pruning algorithm is difficult. To our knowledge, the only analytical result [58] is a function $U(P, M, \mathcal{A}, S)$ that upper bounds the average number of nodes visited by a sphere decoder, for a Rayleigh fading channels with a PAM or QAM input alphabet $\mathcal{A}$ and natural ordering of inputs. Though precise analysis is difficult, the broad influence of the various parameters on the average complexity are summarized below.

Remark 8. The average number of nodes visited by a tree-pruning algorithm, $\overline{V}$, shows the following tendencies.

(i) As the symbol alphabet size $|\mathcal{A}|$ increases, $\overline{V}$ tends to increase.

(ii) As the number of channel inputs M increases, $\overline{V}$ tends to increase.

(iii) As the SNR S increases, $\overline{V}$ tends to decrease.

(iv) As the number of channel outputs P increases, $\overline{V}$ tends to decrease.

(v) Finally, V-BLAST spatial ordering of the channel inputs leads to a lower value of $\overline{V}$, when compared to natural ordering of inputs.

We now proceed to justify the claims in Remark 8. The total number of nodes in the tree, namely $(1 + |\mathcal{A}| + \ldots + |\mathcal{A}|^M)$ increases with the alphabet size $|\mathcal{A}|$. Pruning ensures that not all these nodes are visited. However, it is clear that with more nodes in the tree, the number of nodes visited tends to increase, justifying (i). It is worth pointing out that this increase is typically not very sharp, i.e., even though the total number of nodes increases roughly as $|\mathcal{A}|^M$, $\overline{V}$ typically increases much more slowly as $|\mathcal{A}|$ increases.

As M increases, the total number of nodes in the tree increases exponentially. Even more seriously, the expected value of the cheapest leaf node cost MN_0, and hence the typical value of the threshold, increases with M. Thus, as M increases, not only are there more nodes in the tree, but fewer of them are pruned out. As a result, $\overline{V}$ tends to increase with M, as stated in (ii). As the SNR S increases, the noise variance N_0 decreases, leading to lower thresholds, and hence a lower value of $\overline{V}$. This justifies (iii).

Statements (iv) and (v) relate to the dependence of $\overline{V}$ on the distribution of the effective channel matrix $\mathbf{L}$. Substituting $\mathbf{y} = \mathbf{L}\mathbf{x} + \mathbf{n}$ in (173), we get the expression

$$C(z_1, z_2, \ldots, z_k) = \sum_{i=1}^{k} \left| n_i + \sum_{j=1}^{i-1} l_{ij}(x_j - z_j) \right|^2 \tag{176}$$

for the cost of the node $(z_1, z_2, \ldots, z_k)$ at depth i. Apart from the additive noise terms n_i, we see that every *wrong* node, i.e., every node $(z_1, z_2, \ldots, z_i)$ that deviates from the actual channel-input symbols $(x_1, x_2, \ldots, x_i)$, has a constant bias terms in the cost expression.

As the number of channel outputs P increases, the receiver has more observations of the channel-input and hence higher received signal energy. Equivalently, a greater value of P leads to a greater energy in each term of $\mathbf{L}$, and hence increases the value of the bias

terms in the cost of wrong nodes. Thus, as P increases, wrong nodes tend to have higher cost, and are hence pruned out by tree-pruning algorithms, leading to a reduction in the average number of nodes visited, as claimed in (iv).

Finally, given P, the receiver's choice of the spatial order does not change the total energy in $\mathbf{L}$, but changes the distribution of the total energy among the various entries of $\mathbf{L}$. To see the effect of ordering on node costs, note from (176) that the cost of a node z_1 of depth one is $|n_1 + l_{11}(x_1 - z_1)|^2$. Recall that V-BLAST spatial ordering maximizes the value of l_{11}. Therefore, wrong nodes at depth one, i.e, nodes corresponding to $z_1 \neq x_1$, tend to have higher cost with V-BLAST order than with natural ordering. Further, given l_{11}, V-BLAST ordering next maximizes l_{22}, and hence increases the cost of wrong nodes of depth two. Proceeding thus, it is easy to see that V-BLAST ordering increases the costs of wrong nodes at lower depths, leading to more low-depth nodes being pruned out. Since nodes at lower depths have a larger number of descendants, V-BLAST ordering, in effect, leads to a greater number of nodes being pruned out, and hence a fewer number of nodes visited. Thus, claim (v) holds for any tree-pruning algorithm. For the case of the sphere decoder, the complexity reduction due to V-BLAST ordering was pointed out in [64].

11.7 The Performance of Tree-Pruning Algorithms

In this section, we present simulation results demonstrating the low error rate and low average computational complexity of tree-pruning algorithms. In particular, these results illustrate that tree-pruning algorithms result in dramatically lower error rates than the SC detector, with only a moderate increase in computational burden.

11.7.1 Comparison of the Error Rate of Detection Algorithms

We first compare the word error rates achieved by the various detectors, for the case of an 8-input, 8-output Rayleigh fading channel and a 16-QAM alphabet. Since each of the $M = 8$ transmit symbols carries $\log_2(16) = 4$ bits of information, the data rate is 32 b / s / Hz. The WER achieved by a bounded stack detector for stack size limits of 1, 10 and 25 is plotted against SNR in Fig. 18. Also shown is the WER achieved with unlimited stack size, corresponding to ML detection. For each stack size, WER with natural ordering is shown in dotted lines, and WER with V-BLAST ordering is shown in solid lines.

A stack size limit of one yields the successive cancellation detector, which has the highest error rate. Note that V-BLAST ordering improves the performance of the SC detector significantly, requiring 5 dB less SNR to achieve a WER of 10^{-1}. Roughly, this improvement is because the V-BLAST ordering, which is a greedy ordering algorithm, is ideally suited to the greedy SC detector. For a more precise discussion, see [11]. As the

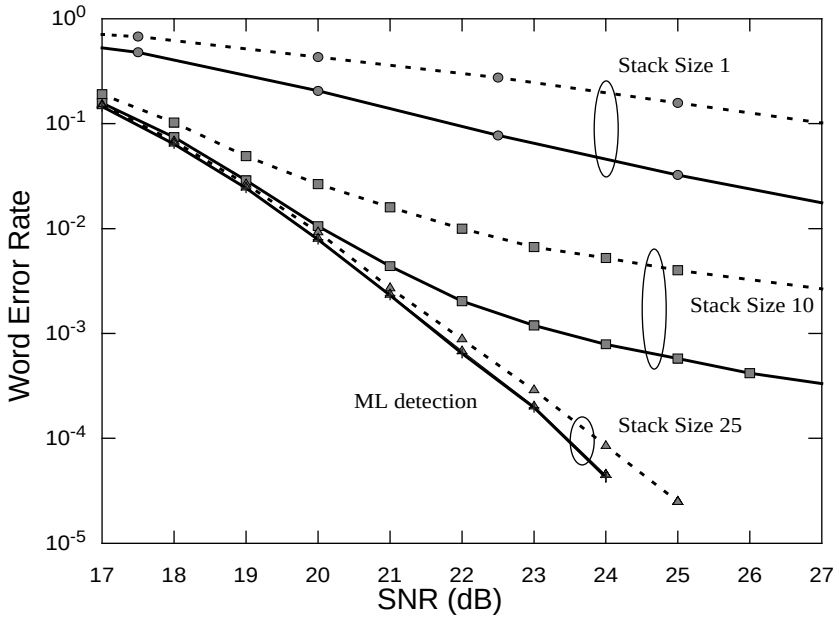

Fig. 18. Variation of bounded stack decoder performance with stack size for an 8-input, 8-output Rayleigh fading channel at 32 b/s/Hz.

stack size limit increases from one to ten, the error rate drops sharply. Remarkably, a stack size of merely 25 nodes achieves almost identical performance to ML detection, which corresponds to unlimited stack size.

To achieve a WER of 10^{-2}, ML detection needs 25 dB less SNR than V-BLAST ordered SC detector. This dramatic performance improvement is primarily because of the difference in diversity orders achieved by the ML and SC detectors. With ML detection, the full raw diversity order of the S/P converter is achieved, as discussed in Chapter 3. In contrast, the SC detector does not harvest the diversity benefit of multiple Rayleigh fading channel outputs, and hence leads to dramatically higher WER at high SNR.

Finally, note that there is only one curve marked ML detection, indicating that the ML detector's WER does not depend on the ordering choice. This is true in general, and is easy to show formally [7]. Intuitively, the SWMF output carries the same probabilistic information about the channel-input, irrespective of the spatial order. Since the ML detector optimally uses all the available information, ordering does not impact the WER of the ML detector. However, we will now see that V-BLAST ordering does reduce the number of nodes visited by a tree-pruning algorithm, as claimed in Remark 8.

11.7.2 Average Computational Complexity of the Sphere Decoder

In this section, we illustrate some of the general remarks about the computational complexity of tree-pruning algorithms with simulation results. For the system in Section 11.7.1, namely an 8-input, 8-output Rayleigh fading channel with a 16-QAM symbol alphabet, the average number of nodes visited by the sphere decoder is plotted

against SNR in Fig. 19. To magnify the differences between the plots, we have plotted the logarithm of the average number of nodes visited to the base $M = 8$. Recall that any tree-pruning algorithm saves computation by a two-fold strategy: first, pruning ensures that only nodes with cost less than threshold are visited; second, the initial threshold C_0 is cleverly chosen as $\alpha MN0$, and scaled by a factor β whenever erasure is detected. To understand the contribution of each of these steps, we have plotted the average number of nodes visited with only pruning, and no clever threshold initialization, i.e, with $C_0 = \infty$. (The threshold is still tightened, whenever a leaf node is reached). As seen from Fig. 19, this step alone is sufficient to make the sphere decoder visit dramatically fewer nodes than the total number of nodes in the tree, namely 4,581,298,449. With clever threshold initialization, i.e., using $\alpha = \beta = 1.5$, the average complexity reduces further, as seen in Fig. 19. Both the above curves showed the average number of node visits with natural ordering of inputs. When V-BLAST ordering is added on to the cleverly initialized sphere decoder, the average number of nodes visited nearly halves.

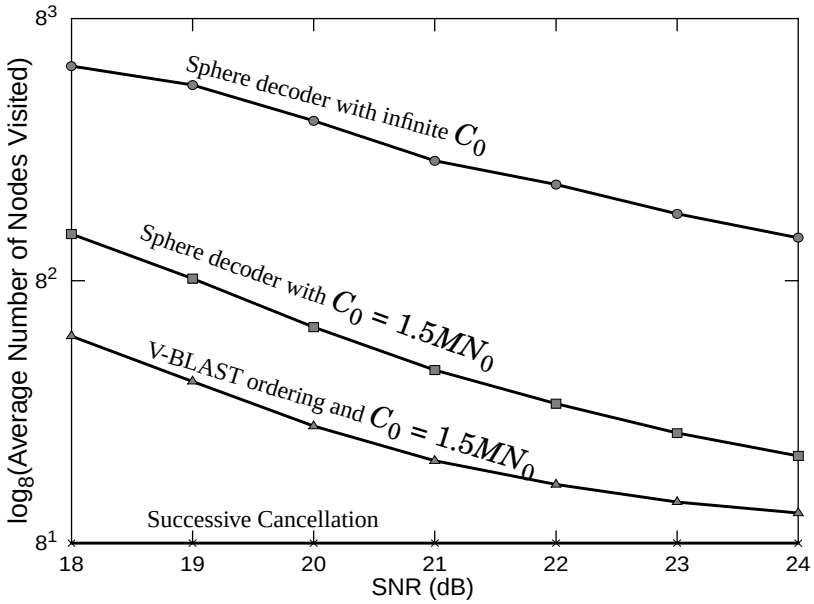

Fig. 19. Average complexity of sphere decoder for an 8-input, 8-output Rayleigh fading channel at 32 b/s/Hz.

Note that for all these cases, the average complexity reduces as the SNR S increases, as expected from (iii) of Remark 8. Remarkably, at SNR = 24 dB, the V-BLAST ordered sphere decoder visits only 10 nodes on the average, merely 25% more than the $M = 8$ nodes visited by the SC detector. In return, we see from Fig. 18 that the sphere decoder achieves ML detection whose WER at 24 dB is around 5×10^{-5}, which is only one thousandth of the WER of nearly 5×10^{-2} achieved the SC detector. Thus, when the SNR is high, tree-pruning algorithms like the sphere decoder are a very attractive alternative to the successive cancellation detection algorithm.

11.7.3 Comparison of the Hybrid and Sphere Decoders

In the last section, we used the sphere decoder as an example to illustrate the low computational complexity of tree-pruning algorithms. In this section, we compare the average number of nodes visited by the sphere decoder and the hybrid decoder.

It is instructive to first compare the movement pattern of the sphere and hybrid decoders. From the algorithms in Fig. 16 and Fig. 17, it is clear that both decoders start from the root node and roughly perform SC detection first, i.e., successively take the cheapest branch forward. Suppose this procedure does lead to the cheapest leaf node. Then, both decoders tighten the threshold to $C_{\min}$. Of course, the two decoders do not know that the cheapest leaf node has already been found, and continue the search. The key observation is that subsequently, both decoders will visit all the nodes in the tree with cost less than $C_{\min}$, before the search terminates. In particular, note that if the SC detector does find the cheapest leaf node, both the hybrid and sphere decoders visit exactly the same number of nodes.

However, the SC procedure at the beginning of hybrid and sphere decoding does not always find the cheapest leaf node. Sometimes, it reaches a leaf node which is not the cheapest one. At other times, it terminates without reaching a leaf node, because all branches of some node get pruned out. In this case, both decoders continue their search, but in different *sections* of the tree. The sphere decoder looks to move forward from the deepest available node, while the hybrid decoder visits a node of depth one, and attempts successive cancellation starting from it. The decoder which finds the cheapest leaf node sooner tightens the threshold to $C_{\min}$, and hence visits fewer nodes subsequently than the other decoder.

Whether the sphere decoder finds the cheapest leaf node sooner than the hybrid decoder depends on the distribution of the channel matrix $\mathbf{L}$. For Rayleigh fading channels, with natural ordering at the receiver, it is well known [43] that the term l_{ii} has a chi-squared distribution of degree $2(P - M + i)$, implying that as i increases, the coefficient l_{ii} tends to increase. Since the magnitude of l_{ii} determines the reliability of stage i, we conclude that early stages of the tree tend to be unreliable, i.e., wrong nodes do not have significantly higher cost than the right node. However, the sphere decoder always searches for the cheapest leaf node by moving forward from the *deepest* available node. In other words, it believes that decisions made in the early stages of the decoding tree are correct, and attempts to make the right decisions for later decisions. Since this assumption runs counter to the actual statistical behavior of the channel, the sphere decoder is ill-suited to the case of unordered Rayleigh fading channels. On the other hand, V-BLAST ordering increases the value of l_{ii} for lower values of i, and makes the early stages more reliable. Hence, the sphere decoder has low computational complexity with V-BLAST ordering.

To sum up, we expect that the sphere decoder should have a significantly higher average complexity than the hybrid decoder, when the receiver uses natural ordering of inputs. With V-BLAST ordering of inputs, the sphere and hybrid decoders should visit roughly the same number of nodes on the average. This prediction is confirmed in Fig. 20, which shows the average number of node visits by the sphere and hybrid decoders for a 16-input, 16-output Rayleigh fading channel with 16-QAM input symbols. With natural ordering of inputs, at low SNR, the hybrid decoder visits about 15-20% fewer nodes on the average, when compared to the sphere decoder. With V-BLAST ordering, the difference in the number of node visits is almost negligible. With either ordering, the difference between the number of node visits is negligible at high SNR.

It is instructive to compare the average number of nodes visited for the present system with corresponding values for the system considered in Section 11.7.2. The former system had 8 channel inputs and outputs with a 16-QAM alphabet. At an SNR of 21 dB, the sphere decoder visits about 11 nodes on the average, as seen from Fig. 19. When the

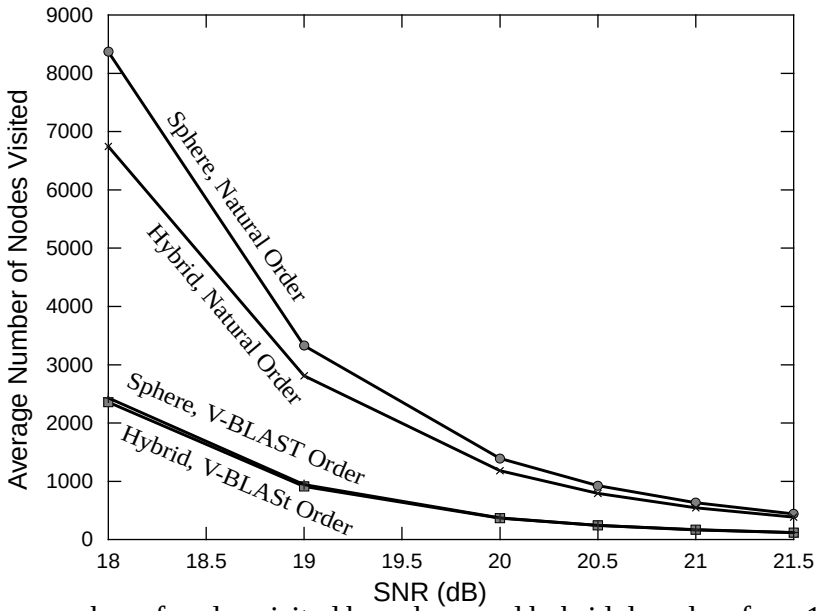

Fig. 20. Average number of nodes visited by sphere and hybrid decoders for a 16-input, 16-output Rayleigh fading channel at 64 b/s/Hz.

number of channel inputs and outputs increases to 16, Fig. 20 shows that the average number of nodes visited increases to more than 100 at the same SNR. Extrapolating this trend, one can see that the sphere decoder (or other tree-pruning algorithms) tend to become impractically complex beyond a few tens of channel inputs.

11.8 Conclusions

When the transmitter consists of a stand-alone space-time code with a discrete input alphabet, the receiver must perform ML detection in order to minimize the error probability. In this chapter, we interpreted ML detection as a tree-search problem, and discussed the class of tree-pruning algorithms, which efficiently solve the tree-search problem. The tree-pruning algorithm that performs depth-first search is identical to the well-known sphere decoder. We also developed the new hybrid search decoder, and its non-ML variant, the bounded stack hybrid search decoder.

The computational burden of tree-pruning algorithms depends on the random channel matrix and noise. Simulation results show that for typical MIMO channel dimensions, the *average* computational complexity of tree-pruning algorithms is in the same order of magnitude as the *fixed* computational complexity of the successive cancellation detector used in the V-BLAST architecture. Thus, tree-pruning algorithms satisfactorily solve the problem of designing optimum, computationally simple receivers for transmitters with stand-alone linear space-time codes.

CHAPTER 12

Soft-Output Decoders for Linear MIMO Channels

In this chapter, we consider the design of receivers for the second transmitter architecture considered in this work, namely the concatenation of an outer finite-field error correction code with an inner space-time code (see Fig. 8). In order to minimize the error probability, the receiver should ideally perform maximum likelihood decoding, treating the concatenation of the outer and inner codes as one super-code. However, for large outer code lengths, the optimum joint ML decoder has prohibitive computational complexity.

A near-optimum alternative is to perform *iterative decoding*, following the pattern of turbo codes [31]. Here, probabilistic information is iteratively exchanged between *soft-output* decoders for the inner space-time code and the outer code respectively, as shown in Fig. 21. Heuristically, it is well known that the estimates produced by the iterative decoder after a few iterations are almost as reliable as those of the impractical joint decoder.

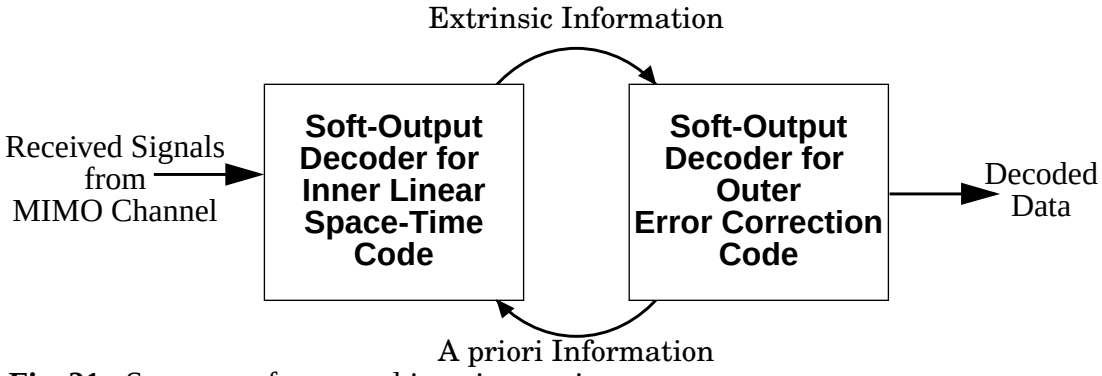

Fig. 21. Structure of a general iterative receiver.

To implement the iterative receiver structure of Fig. 21, soft-output decoders are required for the outer code and inner space-time code. In this work, we have assumed that the outer code is a finite-field code, for which soft-output decoders are well known [32]. In this chapter, we develop soft-output decoders for a general linear space-time code, by a simple extension of the detection algorithms developed in the previous chapter.

12.1 A Precise Problem Statement for Soft-Output MIMO Decoders

The receiver uses the spatial whitened matched filter described in Section 11.1.1 to yield the spatially causal effective channel (166) $\mathbf{y} = \mathbf{L}\mathbf{x} + \mathbf{n}$. Each symbol x_i is obtained by modulating a group of bits from the outer code. More precisely, the outer code produces coded finite-field symbols, which are first broken down to bits, and then interleaved. The interleaved bits are separated into groups containing $|\mathcal{A}|$ bits each. Each group then addresses a look-up table to read off a channel-input symbol belonging to the alphabet $\mathcal{A}$. Thus, each channel-input symbol x_i uniquely corresponds to a set of $|\mathcal{A}|$ coded and interleaved bits, say $b_{i,1}, b_{i,2}, \ldots, b_{i,|\mathcal{A}|}$.

In the detection problem, we were not interested in the bit labels $\{b_{i,j}\}$, but directly in the symbols x_i. However, soft-output MIMO decoders have to exchange probabilistic information about the bits with the soft-output decoder for the outer code. More precisely, the soft-output outer decoder provides the *a priori* log likelihood ratio (LLR) $A_{i,k}$ on each input bit $b_{i,k}$, $k = 1, 2, \ldots, |\mathcal{A}|$. The LLR is just a convenient way of storing the probability distribution of each bit, which is given by

$$p(b_{i,k} = b) = \frac{e^{bA_{i,k}}}{1 + e^{A_{i,k}}} \tag{177}$$

for $b = 0, 1$. In return for the a priori information from the outer decoder, the soft-output inner decoder provides *extrinsic information* to the outer decoder as follows. First, it uses the effective channel output $\mathbf{y}$ and the a priori LLRs to compute the *a posteriori* LLR

$$L_{i,\,k} = \log\left(\frac{p(b_{i,\,k} = 1|\mathbf{y})}{p(b_{i,\,k} = 0|\mathbf{y})}\right) = \log\left(\frac{p(b_{i,\,k} = 1, \mathbf{y})}{p(b_{i,\,k} = 0, \mathbf{y})}\right). \tag{178}$$

(The second equality above is obtained using Bayes' rule.) From the a posteriori LLR, the soft-output decoder computes the extrinsic LLR $E_{i,k} = L_{i,k} - A_{i,k}$, which is then passed on to the outer decoder. The outer decoder then uses the extrinsic information to compute the a priori information for the next iteration, and so on.

12.2 Soft-Output Decoding As a Tree-Search Problem

In this section, we interpret the soft-output decoding problem in terms of the detection tree. The presence of a priori information can be accounted for by a slight change to the previously defined branch and node costs. In the next section, we will extend the detection algorithms of the previous chapter, to obtain soft-output tree-pruning algorithms.

First, we rephrase the soft-output decoding problem, to obtain a convenient node cost function. Using the one-to-one correspondence between input bits and the transmit vector, (178) for the a posteriori LLR becomes

$$L_{i,\,k} = \log\left(\frac{p(b_{i,\,k} = 1, \mathbf{y})}{p(b_{i,\,k} = 0, \mathbf{y})}\right) = \log\left(\frac{\sum_{\mathbf{z} \in \mathcal{X},\, b'_{i,\,k} = 1} p(\mathbf{y}, \mathbf{z})}{\sum_{\mathbf{z} \in \mathcal{X},\, b'_{i,\,k} = 0} p(\mathbf{y}, \mathbf{z})}\right). \tag{179}$$

To compute (179) rigorously, the joint probability $p(\mathbf{y}, \mathbf{z})$ has to be computed for each valid transmit vector $\mathbf{z}$ and added to the numerator or denominator, depending on the bit label $b'_{i,k}$ corresponding to $\mathbf{z}$. In order to avoid the daunting computational requirements

of this task, the log-max approximation is usually made [32], replacing the sum by the maximum of the summands. Thus, we get

$$L_{i,k} \approx \log \frac{\left(\max_{\mathbf{z} \in \mathcal{X},\, b'_{i,k} = 1} \right) p(\mathbf{y}, \mathbf{z})}{\left(\max_{\mathbf{z} \in \mathcal{X},\, b'_{i,k} = 0} \right) p(\mathbf{y}, \mathbf{z})} \tag{180}$$

$$= \left(\max_{\mathbf{z} \in \mathcal{X},\, b'_{i,k} = 1} \right) \log p(\mathbf{y}, \mathbf{z}) - \left(\max_{\mathbf{z} \in \mathcal{X},\, b'_{i,k} = 0} \right) \log p(\mathbf{y}, \mathbf{z}). \tag{181}$$

To compute the approximate LLR (181) for each bit $b_{i,k}$, one needs to find just two vectors with different values of $b_{i,k}$, with the maximum value of log $p(\mathbf{y}, \mathbf{z})$. Now, the function $-\log p(\mathbf{y}, \mathbf{z})$ can be thought of as a cost function associated with $\mathbf{z}$. Decomposing it as $-\log p(\mathbf{y} \mid \mathbf{z}) p(\mathbf{z})$ and substituting for the conditional probability $p(\mathbf{y} \mid \mathbf{z})$ from (169),

$$-\log p(\mathbf{y}, \mathbf{z}) = -\log p(\mathbf{y} \mid \mathbf{z}) - \log p(\mathbf{z}) \tag{182}$$

$$= N \log(\pi N 0) + \frac{\|\mathbf{y} - \mathbf{L}\mathbf{z}\|^2}{N_0} + \sum_{i=1}^{N} \sum_{k=1}^{K} \left(-\log p(b'_{i,k}) \right). \tag{183}$$

The first term in (183) independent of $\mathbf{z}$ and is cancelled out while calculating the difference in (181). So, it can be ignored. The second term is just the detection problem's cost function $J(\mathbf{z})$ divided by the noise energy N_0. The last term is the a priori cost of the bit labels $\{b'_{i,k}\}$ associated with the transmit vector $\mathbf{z}$. From (177), we get

$$-\log p(b'_{i,k}) = -A_{i,k} b'_{i,k} - \log(1 + e^{A_{i,k}}). \tag{184}$$

The second term in (184) can be neglected without affecting the difference (181). However, the first term $-A_{i,k} b'_{i,k}$ alone could potentially be negative, when the LLR $A_{i,k}$ is positive and $b'_{i,k} = 1$. In this case, to avoid negative a priori bit cost, we just add $A_{i,k}$ to the cost of both bit labels 0 and 1. Thus, we get the non-negative a priori bit cost function

$$C(b'_{i,k}) = -A_{i,k}b'_{i,k} + \frac{|A_{i,k}| + A_{i,k}}{2} = \frac{|A_{i,k}| + (-1)^{b'_{i,k}}A_{i,k}}{2}. \tag{185}$$

Intuitively, $C(b'_{i,k})$ can be thought of as the cost of disagreeing with the a priori information. For example, if $A_{i,k} > 0$, we have a priori information that $b_{i,k} = 1$ with certainty $A_{i,k}$. Correspondingly, the cost $C(1)$ is 0, whereas the opposite decision has cost $C(0) = A_{i,k}$.

Substituting the a priori bit cost function $C(b'_{i,k})$ back in (182), we get the cost function for every valid channel-input vector $\mathbf{z} \in \mathcal{X}$, given by

$$K(\mathbf{z}) = \frac{\|\mathbf{y} - \mathbf{L}\mathbf{z}\|^2}{N_0} + \sum_{i=1}^{N}\sum_{k=1}^{K} C(b'_{i,k}). \tag{186}$$

Up to a constant independent of $\mathbf{z}$, $K(\mathbf{z})$ is equal to $-\log p(\mathbf{y}, \mathbf{z})$. Substituting in (180), we see that the task of the log-max soft-output decoder is to obtain the extrinsic information

$$E_{i,k} = L_{i,k} - A_{i,k} \approx \left(\min_{\mathbf{z} \in \mathcal{X}, b_{i,k}=0}\right) K(\mathbf{z}) - \left(\min_{\mathbf{z} \in \mathcal{X}, b_{i,k}=1}\right) K(\mathbf{z}) - A_{i,k} \tag{187}$$

for all symbols $i = 1, 2, \ldots, M$ and bit indices $k = 1, 2, \ldots, |\mathcal{A}|$ of each symbol.

We have now rephrased the soft-output decoding problem in the form (187). To proceed, we note that the new cost function $K(\mathbf{z})$ breaks up into non-negative branch costs on the detection tree, namely

$$B'(z_1, z_2, \ldots, z_i) = \frac{1}{N_0}\left|y_i - \sum_{j=1}^{i-1} l_{ij}z_j\right|^2 + \sum_{k=1}^{K} C(b'_{i,k}). \tag{188}$$

Note that the new branch costs are similar to the branch costs defined for the detection problem. In fact, the first term is just a scaled version of the detection problem's branch cost (172), and the second term accounts for the a priori information. Similar to the

detection problem, the branch costs are again non-negative. Further, these branch costs can be used to define node costs, as in (173)(174).

Note that the cost of the leaf node corresponding to a channel-input vector $\mathbf{z} \in \mathcal{X}$ is $K(\mathbf{z})$. Thus, to evaluate the soft-output (187) for a bit $b_{i,k}$, the log-max soft-output decoder should find the two cheapest leaf nodes which differ in the label $b_{i,k}$. Clearly, the cheapest leaf node in the tree is always one of the candidates, and it determines the sign of the LLR $L_{i,k}$. For example, if the bit label $b^*_{i,k}$ of the cheapest leaf node is zero, the first term in (187) contains the cheapest leaf node cost and is less than the second term, so that their difference $L_{i,k}$ is negative. In general, the sign of $L_{i,k}$ is $(2b^*_{i,k} - 1)$. The magnitude of $L_{i,k}$ is determined by the cost of the *competing* node for each bit, namely the cheapest leaf node whose bit label is different from $b^*_{i,k}$. Thus, to implement soft-output decoding, a tree search algorithm should determine not only the cheapest leaf node, but also the cheapest competitor for every bit.

12.3 Extension of MIMO detectors to Obtain Lists

The tree-pruning algorithms of the previous chapter already find the cheapest leaf node in the tree. One expects that they can be easily extended to generate a list of cheap leaf nodes for soft-output generation. Such an extension was proposed for the sphere decoder [33], yielding the list sphere decoder. In this section, we review the extension of [33] and propose an improved extension procedure, which is applicable to all tree-pruning algorithms. We will see that only the threshold initialization and threshold updation at leaf nodes need to be changed for a detector to produce lists. For the sake of simplicity, we will specifically discuss extension of the sphere decoder.

The list sphere decoder [33] was developed for the case of no a priori information, i.e., $A_{i,k} = 0$. It obtains a list $\mathcal{L}$ of the N_{cand} (or fewer) cheapest leaf nodes with cost less than the initial threshold $C_0 = \alpha MN_0$. Soft outputs are then calculated by taking the minima in (187) over the list $\mathcal{L}$ instead of the entire transmit alphabet $\mathcal{X}$. Sometimes, all the leaf nodes in $\mathcal{L}$ might have the same value of some bit $b_{i,k}$. Consequently, one of the minima cannot be computed. In principle, one can iteratively increase C_0 and/or N_{cand} till the list $\mathcal{L}$ is large enough to furnish two cheap competitors for every bit. Instead, whenever all nodes in $\mathcal{L}$ have the same bit label $b^*_{i,k}$, the list sphere decoder just outputs an *approximate* extrinsic LLR of $L_{\max}(2b^*_{i,k} - 1)$, where $L_{\max}$ is some pre-determined value.

We already saw that the competitor node only determines the magnitude of the extrinsic information. What the approximation implies is that in the absence of a competitor, the list sphere decoder guesses the magnitude to be $L_{\max}$. If the initial threshold C_0 and the list size N_{cand} are large enough, two competitors are found for most bits, and the approximation needs to be done infrequently. Further, it does not significantly affect performance because most fixed point implementations of iterative decoding anyway clip the magnitudes of the a priori and extrinsic LLRs in order to avoid overflows or underflows. Clipping also helps to avoid the chaotic dynamics and sudden bursts of errors, to which iterative decoding of finite length codes is prone [65]. If $L_{\max}$ is be chosen to be the clipping value, the approximate list sphere decoder produces exactly the same output as an ideal soft-output sphere decoder (which expands threshold to obtain competitors for every bit) after clipping.

In [33], the list sphere decoder was implemented by a change to the threshold updation procedure of the original sphere decoder. Whenever a leaf node is reached, we first check the size of the list $\mathcal{L}$. If $\mathcal{L}$ has fewer than N_{cand} candidates, the new leaf node is added to $\mathcal{L}$. Otherwise, the current leaf node replaces the costliest leaf node in $\mathcal{L}$ only if it has a lower cost. The threshold is never tightened.

We propose two modifications to the threshold updation procedure proposed in [33]. Firstly, the threshold can clearly be tightened to the maximum leaf node cost in $\mathcal{L}$, if $\mathcal{L}$ is full. This is because any subsequent leaf node with a cost greater than this threshold will not be added to the list anyway, and hence one can avoid visiting all such leaf nodes without affecting the final output.

Secondly, suppose $K_{\min}$ is the cost of the cheapest leaf node in $\mathcal{L}$, and $A_{\max}$ and $L_{\max}$ are respectively the maximum magnitude of the a priori LLRs $\{A_{i,k}\}$, and the clipping value for the output extrinsic LLR. Then, one can tighten the threshold to $K_{\min} + A_{\max} + L_{\max}$, if the latter quantity is lower than the current threshold. The rationale behind this is as follows. Suppose a leaf node L' in the list has cost $K > K_{\min} + A_{\max} + L_{\max}$. If L' is not one of the minima used in (187) to calculate the extrinsic information, we anyway do not need it in the list. Even if it is one of the minima, we see that the resulting extrinsic information would satisfy

$$|E_{i,k}| = |\pm (K - K_{\min}) - A_{i,k}| \geq |(K - K_{\min})| - |A_{i,k}| \geq (K - K_{\min} - A_{\max}) > L_{\max}. \quad (189)$$

Since $|E_{i,k}| > L_{\max}$, clipping would just reduce $E_{i,k}$ to $L_{\max}(2b^*_{i,k} - 1)$. Now, if L' had not been added on the list, the closest competitor would not have been found, but the list sphere decoder's built in assumption would still have produced the same output $L_{\max}(2b^*_{i,k} - 1)$ as if L' had been there. Thus, leaf nodes with cost greater than $K_{\min}$ +

$A_{\max} + L_{\max}$ do not change the ultimate output of the list sphere decoder, and can be discarded by tightening the threshold to $K_{\min} + A_{\max} + L_{\max}$. To sum up, we propose the following leaf node processing.

```
Leaf Node Processing if (CurrentNode is a leaf node)
                        if (ℒ has fewer than N_cand entries)
                            add CurrentNode to ℒ
                        else
                            if (CurentNode.Cost < Maximum cost in ℒ)
                                Replace costlier leaf node in ℒ by CurrentNode
                                if (Threshold > Maximum cost in ℒ)
                                    Threshold = Maximum cost in ℒ
                            endif
                        endifelse
                        A_max = Maximum value of |A_{i,k}|
                        L_max = Cutoff value of output extrinsic information
                        if (Threshold > (CurrentNode.Cost + A_max + L_max))
                            Threshold = CurrentNode.Cost + A_max + L_max
                        endif
                    endif
```

Fig. 22. Leaf node updation step that extends a detection algorithm to produce lists.

The list size N_{cand} and the cutoff value $L_{\max}$ should be chosen to optimally tradeoff performance and complexity. Making either of them large would imply large lists, and intensive computation, but would reduce the frequency of not finding the closest competitor, and hence enable the decoder to produce nearly log-max outputs. The list size N_{cand} should increase with the data rate of operation. Values of the order of 100 were suggested in [33]. We have found that $L_{\max}$ of around 5.0 to 10.0 works well.

In addition to threshold updation, the threshold initialization of the sphere decoder should also be changed, to enable the handling of a priori information. For the detection problem, the initial threshold C_0 was chosen as $\alpha MN0$ to reflect the average cost of the actual transmit vector. Now, the transmit vector could incur an additional a priori cost, if some of the a priori LLRs are in error. To account for this, we propose to add an additional term proportional to the average a priori LLR magnitude A (the mean of $|A_{i,k}|$ over all the

NK bits). More precisely, we propose to use $C0 = \alpha' MN_0 + \delta MA$. The constant α' should be larger than the α for the detector, because we wish to visit more leaf nodes than just the cheapest one.The factor δ represents the estimated number of a priori LLR errors per branch, since we are allowing for a total a priori cost of δNA over a path of N branches. Since a priori information comes for a reliable outer code, δ should be relatively small. Heuristically, we recommend $\alpha' \approx 2.5$ and $\delta \approx 0.2$.

With the new threshold initialization and the updation steps mentioned above, all the tree-pruning detectors can be extended to provide soft outputs. No other change is required.

12.4 Simulation Results

We now present simulation results to confirm the efficiency of tree-pruning soft-output decoders, and the complexity reduction obtained by using the proposed threshold updation procedure. A rate-$1/2$ (4800, 2400) binary turbo code is used as the outer code. The constituent codes are both punctured $\left[1, \frac{1+D+D^4}{1+D+D^2+D^3+D^4}\right]$ convolutional codes, and a spread-20 random interleaver was used in the turbo code. The coded bits are interleaved using a spread-26 random interleaver, assembled into 4-bit symbols, and modulated to obtain 16-QAM complex symbols. These are split into four streams and transmitted over a 4-input, 4-output Rayleigh fading channel, which remains constant over the entire frame, lasting 300 signaling intervals. Thus, the data rate is $2400/300 = 8$ b / s / Hz. At the receiver, the inner soft-output decoder does a maximum of 5 iterations with the outer turbo decoder. Iterations were stopped when zero bit errors were detected. The turbo code itself does four internal iterations for every iteration with the inner soft-output MIMO decoders.

We compare three different soft-output decoding algorithms, namely list-generating extensions of a sphere decoder and two bounded stack hybrid decoders of size limits thirty nodes and ten nodes respectively. All these generated lists of maximum size $N_{\text{cand}} = 256$, and had clipping values of $L_{\max} = 5.0$. The threshold was initialized using $\alpha' = 2.5$ and $\delta = 0.2$. The average number of nodes visited per signaling interval is plotted, i.e., this is the sum of node visits over all the iterations with the outer code.

In Fig. 23, the plot on the left shows the frame error rate achieved by the different soft-output decoding algorithms at different SNR. Recall that limiting the stack size limits the memory requirement of the hybrid decoder, but leads to an increase in error rate. However, we saw in Section 11.7.1 that even for small stack size limits, the error rate penalty when compared to ML detection is small. Similar results hold for the soft-output extension of the bounded stack decoder, too. With a stack size limit of just 30 nodes, the bounded stack

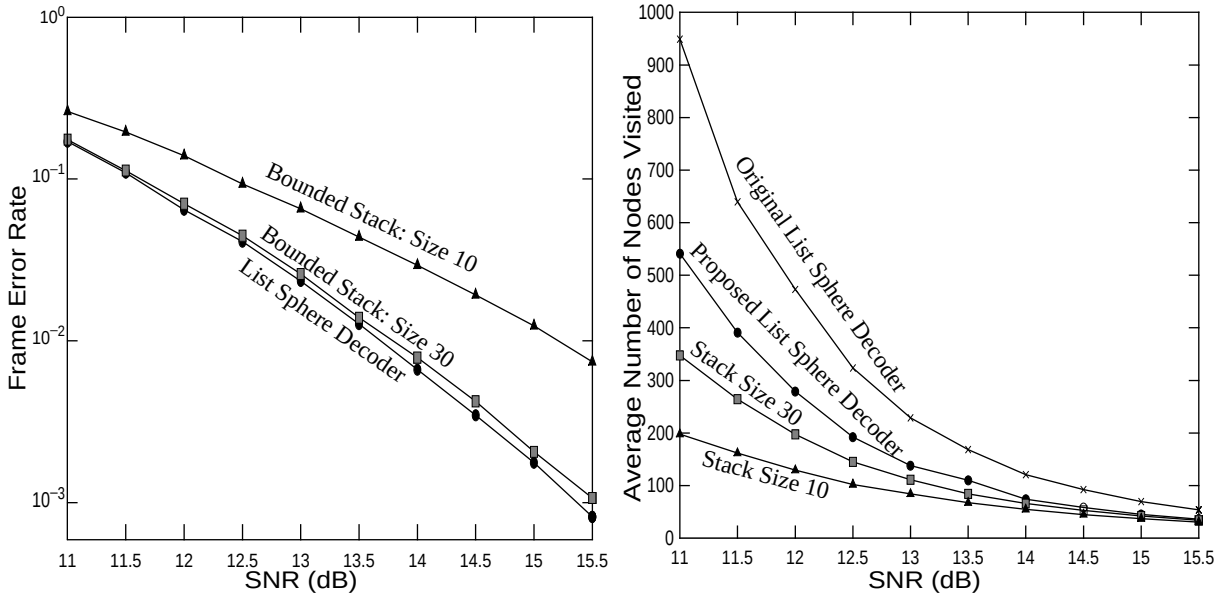

Fig. 23. Comparing FER and average number of nodes visited by different soft-output decoders, for a 4×4 Rayleigh fading channel, with a rate 1/2 turbo outer code.

hybrid decoder approaches the performance of the list sphere decoder within 0.05 dB. However, when the stack size limit of 10 nodes, the performance loss widens to about 2.0 dB at a frame error rate of 10^{-2}.

The computational complexity of the decoders is shown on the right of Fig. 23. Note that the proposed changes to the original list sphere decoder reduce the number of nodes visited by around 40% for a wide range of SNR. Of course, the actual lists generated are the same for both updations, and hence the FERs are the same for both sphere decoders on the left plot. As expected, the bounded stack decoders have lower average complexity than the list sphere decoder, in return for their higher error rate.

12.5 Conclusions

When the transmitter contains a concatenation of an outer code with a linear inner space-time code, iterative receivers achieve near-optimum performance with acceptable complexity. To implement an iterative receiver, soft-output decoders for the effective channel of the linear space-time code are required. We developed soft-output decoders using the detection tree introduced in Chapter 11, after correcting the branch costs to include a priori information. Just as MIMO detection amounts to the search for the cheapest leaf node in the detection tree, soft-output decoding amounts to generating a list of the cheapest leaf nodes. In this chapter, we extended the tree-pruning detection algorithms of the previous chapter to generate lists of leaf nodes, which are used to generated soft outputs. Simulation results presented here show that the algorithms presented here offer a low-complexity implementation of the near-optimum receiver for a concatenated transmitter.

CHAPTER 13

Conclusions and Future Work

In this work, we studied the design of near-optimum transmitters and low-complexity receivers for communication across a linear, quasi-static frequency-flat Rayleigh fading channel with t inputs and r outputs.

13.1 Contributions to Transmitter Design

On the transmitter side, we restricted attention to linear space-time codes. In Chapter 2, the channel model and the encoding process of linear space-time codes were described. A linear space-time code is completely specified by its encoding matrix. Designing such a code amounts to choosing its encoding matrix to optimize some performance metric that reflects the goodness of a space-time code. The choice of meaningful performance metrics depends on the role of the space-time code in the overall transmitter architecture. Two such architectures were considered. In the first architecture, the space-time code is a stand-alone code, i.e., its inputs are uncoded, and independent from one space-time code block to another. The second architecture contains a powerful outer code concatenated with the space-time inner code. Because of the outer code, the inputs to the space-time inner code are dependent from block to block. Good stand-alone space-time codes are not necessarily good inner codes, and vice versa.

Chapters 3-6 addressed the design of good stand-alone space-time codes with inputs drawn from a discrete alphabet. Two broad parameters that determine the goodness of stand-alone space-time codes are the rate and raw diversity order. The rate of a space-time code measures the amount of redundancy introduced by the space-time code, and the raw diversity order measures the effectiveness of the redundancy. High rate and high raw diversity order are both desirable. In particular, it is desirable to use linear space-time codes with full raw diversity order tr and rate greater than or equal to full rate $\min(t, r)$.

Chapter 4 contains the first original contribution of this work. Here, a random code selection argument was used to show that full raw diversity order is easy to achieve. More precisely, almost any linear space-time code whose encoding matrix has orthonormal columns satisfies the rank rule for optimal raw diversity order. In particular, space-time codes with both full rate and full raw diversity order are aplenty. However, achieving full raw diversity does not guarantee that a space-time code achieves the minimum error rate possible, for a given data rate. Intuitively, one can think of rate and raw diversity order as *broad* or *coarse* performance indicators of a stand-alone space-time code. In Chapters 5 and 6, we addressed the optimization of a *precise* or *fine* performance metric, namely the union bound on the word error rate of a space-time code. The goal was to choose an encoding matrix with minimum union bound, given the encoding matrix dimensions, SNR and input alphabet (hence data rate).

In Chapter 5, we developed analytical tools to find encoding matrices with orthonormal columns that minimize the union bound. However, these tools work only for certain matrix dimensions. In particular, they can be used to optimize the union bound only for codes with either low rate or low raw diversity order.

In order to obtain optimum full rate, full raw diversity space-time codes, we developed approximate numerical optimization techniques in Chapter 6. The underlying idea is to treat encoding matrix design as a constrained optimization problem, which can be solved by adapting the gradient descent algorithm. Simulation results show that numerically optimized encoding matrices achieve significantly lower word error rates than other unoptimized codes with the same rate and raw diversity order. The one major drawback of numerical optimization is its computational complexity. We heuristically developed methods to simplify and accelerate the optimization process. In particular, for codes of length two and rate equal to the number of transmit antennas, we presented a special structure for encoding matrices. By restricting the search to encoding matrices with this structure, one can quickly find near-optimum encoding matrices.

To sum up the discussion of stand-alone space-time codes, we have shown that the optimization of the raw diversity order is trivial, and presented numerical techniques for optimizing the union bound.

Chapter 7 begins our discussion of the concatenated transmitter architecture, by showing that the raw diversity order and union bound do not accurately reflect the goodness of space-time inner codes. Instead, we use information theoretic metrics to evaluate space-time inner codes, implicitly assuming that the best possible (infinite length) outer code is used. The strategy is to first apply information theory to obtain the performance of the best possible code over the Rayleigh fading MIMO channel of interest. Then, we analyze the performance of a given space-time inner code operating over the same Rayleigh fading channel, again assuming the best outer code. Comparing the two, we estimate how close the given space-time inner code is to the best possible code.

In Chapter 8, information theoretic analysis is applied to study the highest possible data rates and least possible error probabilities of communication across a Rayleigh fading MIMO channel. The outage probability and outage capacity were introduced. By analyzing these two metrics at high SNR, the diversity and multiplexing orders were computed. The diversity order of a t-input, r-output Rayleigh fading channel is known to be tr. We proved that the multiplexing order is $\min(t, r)$ not only for Rayleigh fading channels, but for all fading channels whose channel matrix is full rank with probability one. The outage capacity asymptote was shown to contain information about both the diversity and multiplexing aspects of the channel. It is less comprehensive than the multiplexing-diversity tradeoff curve [36], but easier to compute and analyze.

Chapter 9 repeats the information theoretic analysis for the effective channel formed by the combination of a space-time code and the Rayleigh fading channel. The multiplexing order of a rate R linear space-time code operating over a t-input, r-output Rayleigh fading channel is at most $\min(t, r, R)$. We also proved that this upper bound on multiplexing order is achieved by most practical linear space-time codes. While the rate of a linear space-time code is an *upper* bound on its multiplexing order, the raw diversity order is a *lower* bound on the *achievable* diversity order in the presence of an outer code. In particular, the S/P converter, whose raw diversity order is only r, has full achievable diversity order tr.

Chapter 10 used the information theoretic analyses in the previous chapters to develop broad design rules for space-time inner codes, in the presence of a powerful outer code. Space-time inner codes must have a rate of at least full rate of $\min(t, r)$. Otherwise, their low rate would result in low multiplexing order, and hence a loss of a significant fraction

of fading channel's capacity at high SNR. On the other hand, space-time inner codes need not have high raw diversity order, since the outer code can achieve higher diversity by coding across multiple space-time code blocks.

Just as rate and raw diversity order serve as coarse design metrics for stand-alone space-time codes, the multiplexing and achievable diversity orders serve as coarse design metrics for space-time inner codes. The analysis in Chapter 10 effectively solves the coarse design problem. The outage probability offers an upper bound on achievable error probability, and is thus analogous to the union bound for stand-alone space-time codes. Consequently, it can be used as fine optimization metric to find the optimum encoding matrix for space-time codes. We pointed out that finding the encoding matrix with minimum outage probability for general encoding matrix dimensions is an open problem. However, we used Telatar's conjecture to argue that S/P converter is one space-time code that achieves minimum outage probability. This leads us to argue in favor of a simple concatenated architecture consisting of a power outer code, with the S/P converter as inner code. This architecture is simple to implement and optimal when the outer code has infinite length.

13.2 Future Work on Transmitter Design

For stand-alone space-time codes, the advent of the full-rate, full raw diversity linear complex field (LCF) codes in [19][20][22], and our proof of the commonness of such codes, has solved the coarse design problem. Thus, we now have linear space-time codes which are roughly good at near-infinite SNR. But the fine design problem, of finding

codes that actually minimize the error rate at finite SNR, is open. Approximate optimization can be done numerically as indicated here, but there is the need for a more analytically rigorous and/or computationally simple solution.

At the other end of the spectrum are space-time inner codes concatenated with infinite-length, optimally designed outer codes. Information theoretic analysis of fading channels and space-time codes is considerably mature. The notion of a trade-off between multiplexing and diversity gains in fading channels, introduced in [36], has opened up the new problem of computing the trade-off curve both for general MIMO fading channels, and for space-time codes operating over these channels.

In our opinion, the more important open problems are practical. As indicated by the example of the S/P converter, space-time codes that are optimum with an outer code could be grossly sub-optimum in the absence of an outer code. Actual outer codes have finite length and lie somewhere between the extremes of ideal outer code and no outer code. Obtaining optimum space-time inner codes and corresponding outer codes for the practical outer code lengths is a challenging open problem, with immense practical significance. Some recent work on this problem can be found in [49][50].

13.3 Contributions to Receiver Design

Given the transmitter structure, the optimum receiver is the one that estimates the transmitted data with minimum probability of error. We discussed optimum receiver design separately for the two transmitter structures under consideration.

In Chapter 11, we considered the case when the transmitter uses a stand-alone linear space-time code. In this case, the optimum receiver performs maximum likelihood (ML) detection, which can be implemented efficiently using tree-pruning algorithms. The basic ideas of tree-pruning algorithms are already present in the literature [7][23][24], and have in fact been used to develop the sphere decoder (see, for example, [30]) for MIMO detection. Our contribution was to explicitly state the rules of all tree-pruning algorithms. We pointed out the sphere decoder is a depth-first tree-pruning algorithm. We also developed a new tree-pruning algorithm, namely the hybrid decoder, whose strategy is a mix of the depth-first and breadth-first search strategies. The hybrid decoder is more suited to high-speed parallel implementation, but uses up more memory. Limiting the memory of the hybrid decoder yields a new sub-optimum algorithm that flexibly trades off memory for error rate.

When the transmitter contains an outer code concatenated with the inner linear space-time code, iterative receivers achieve near-optimum performance with realistic complexity. While soft-output decoders for (typically algebraic) outer codes are well known, soft-output decoders for the inner space-time code are still being developed. In [33], the hard-output sphere decoder was extended to obtain the soft-output *list* sphere decoder. We proposed some changes to the extension procedure. The proposed changes reduce the computational burden, without changing the output of the list sphere decoder. Further, we generalized the extension procedure to obtain soft-output versions of all the tree-pruning detection algorithms that were developed for stand-alone space-time codes.

13.4 Future Work on Receiver Design

Tree-pruning algorithms solve the detection problem and soft-output decoding problem with moderately low computational complexity. Obtaining even more efficient algorithms, or proving that none exist, is an open problem. Even if the simplest possible algorithms are found, their complexity could be unacceptable for some applications. For such applications, it is an open problem to develop receivers that minimize the error rate, while meeting the given complexity constraints.

More interestingly, in this work, the transmitter was designed assuming optimum decoding at the receiver. If complexity constraints prevent the use of the optimum receiver, it is beneficial to redesign the transmitter to best suit the actual sub-optimum receiver. Recent work developed some near-optimum transmitters [67][68][69] for the case where the receiver employs the sub-optimum successive cancellation decoder. However, more extensive transmitter optimization for the successive cancellation decoder and other sub-optimum decoders is an interesting area of future research.

APPENDIX A

Derivation of Pairwise Error Probability of a Space-Time Code

Here, we derive an expression for the pairwise error probability $\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}'))$. Recall that $\mathcal{E}_{\mathbf{u}}(\mathbf{u}')$ occurs if $\mathbf{u}'$ is more likely than $\mathbf{u}$, i.e., if $p_{\mathcal{Y}|\mathcal{U}}(\mathbf{Y} \mid \mathbf{u}') > p_{\mathcal{Y}|\mathcal{U}}(\mathbf{Y}/\mathbf{u})$. From (22), we see that this occurs only if and only if $\|\mathbf{Y} - \mathbf{H}\mathbf{X}'\|_{\mathcal{F}} < \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_{\mathcal{F}}$, where $\mathbf{X}$ and $\mathbf{X}'$ are the transmit matrices corresponding to $\mathbf{u}$ and $\mathbf{u}'$ respectively. Note that $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}$, so the event $\mathcal{E}_{\mathbf{u}}(\mathbf{u}')$ depends on both the Rayleigh fading channel matrix $\mathbf{H}$, and the noise matrix $\mathbf{N}$. For a given channel matrix $\mathbf{H}$, the probability that $\|\mathbf{Y} - \mathbf{H}\mathbf{X}'\|_{\mathcal{F}} < \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_{\mathcal{F}}$ can be obtained using standard AWGN analysis techniques, giving

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}') \mid \mathbf{H}) = \mathrm{Q}\left(\frac{\|\mathbf{H}(\mathbf{X} - \mathbf{X}')\|_{\mathcal{F}}}{\sqrt{2N_0}}\right), \tag{A-1}$$

where $\mathrm{Q}(.)$ denotes the standard Gaussian tail function. To get the PEP, we just need to average the above over the random Rayleigh fading channel matrix $\mathbf{H}$, i.e.,

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) = \int_{\mathcal{H}} \Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}') \mid \mathbf{H})\, p_{\mathcal{H}}(\mathbf{H}) d\mathbf{H}. \tag{A-2}$$

$p_{\mathcal{H}}(\mathbf{H})$ is the probability density function of the $r \times t$ Rayleigh fading matrix $\mathbf{H}$, given by $p_{\mathcal{H}}(\mathbf{H}) = \frac{1}{\pi^{tr}} \exp(-(\|\mathbf{H}\|_{\mathcal{F}}{}^2)$. To facilitate the computation of (A-2), we use the Chernoff bound $\mathrm{Q}(x) \le \exp(-x^2/2)$ while substituting (A-1) in (A-2). Thus, we get

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) \le \frac{1}{\pi^{tr}} \int_{\mathcal{H}} \exp - \left(\frac{\|\mathbf{H}(\mathbf{X}-\mathbf{X}')\|_{\mathcal{F}}^{2}}{4N_0} + \|\mathbf{H}\|_{\mathcal{F}}^{2} \right) d\mathbf{H}. \tag{A-3}$$

In order to simplify the above expression, we first expand the squared Frobenius norm as

$$\frac{\|\mathbf{H}(\mathbf{X}-\mathbf{X}')\|_{\mathcal{F}}^{2}}{4N_0} + \|\mathbf{H}\|_{\mathcal{F}}^{2} = \mathrm{tr}\left(\frac{1}{4N_0}\mathbf{H}(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^{*}\mathbf{H}^{*} + \mathbf{H}\mathbf{H}^{*}\right). \tag{A-4}$$

Then, defining the $t \times t$ matrix

$$\mathbf{R} = \mathbf{I}_t + \frac{(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^{*}}{4N_0}, \tag{A-5}$$

(A-4) becomes

$$\frac{\|\mathbf{H}(\mathbf{X}-\mathbf{X}')\|_{\mathcal{F}}^{2}}{4N_0} + \|\mathbf{H}\|_{\mathcal{F}}^{2} = \mathrm{tr}(\mathbf{H}\mathbf{R}\mathbf{H}^{*}) = \sum_{i=1}^{r} \mathbf{g}_i^{*}\mathbf{R}\mathbf{g}_i, \tag{A-6}$$

where $\mathbf{g}_i$ denotes the i^{th} column of $\mathbf{H}^{*}$. Substituting (A-6) into (A-3), we get

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) \le \frac{1}{\pi^{tr}} \int_{\mathcal{H}} \exp\left(-\sum_{i=1}^{r} \mathbf{g}_i^{*}\mathbf{R}\mathbf{g}_i\right) d\mathbf{g}_1 d\mathbf{g}_2 \ldots d\mathbf{g}_r. \tag{A-7}$$

The vectors $\{\mathbf{g}_i\}$ are independent and identically distributed, hence the integral splits up as

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) \le \frac{1}{\pi^{tr}} \left(\int_{\mathcal{G}} \exp(-\mathbf{g}^{*}\mathbf{R}\mathbf{g}) d\mathbf{g} \right)^{r}. \tag{A-8}$$

Further, each $\mathbf{g}_i$ is a $t \times 1$ with unit-variance, zero-mean complex Gaussian entries, and so the integral above is a standard integral which evaluates to [4]

$$\int_{\mathcal{G}} \exp(-\mathbf{g}^{*}\mathbf{R}\mathbf{g}) d\mathbf{g} = \frac{\pi}{\det(\mathbf{R})}. \tag{A-9}$$

Substituting (A-9) and (A-5) in (A-8), we get the desired expression

$$\Pr(\mathcal{E}_{\mathbf{u}}(\mathbf{u}')) \le \det.\left(\mathbf{I}_t + \frac{(\mathbf{X}-\mathbf{X}')(\mathbf{X}-\mathbf{X}')^{*}}{4N_0} \right)^{r}. \tag{A-10}$$

APPENDIX B

Computing the Derivative of the Union Bound

Here, we compute the derivative of one representative term in the union bound (87) with respect to the elements m_{ij} of the $Nt \times K$ complex matrix $\mathbf{M}$. For the purposes of this appendix alone, it will prove useful to use C/C++ indexing of rows and columns, starting from 0 instead of 1.

Define the $t \times N$ matrix $\mathbf{D}' = \mathrm{mat}(\mathbf{Md})$. Now, defining the $t \times t$ matrix

$$\mathbf{P}' = \mathbf{I}_t + \frac{NS}{4K}\mathbf{D}'\mathbf{D}'^{*}. \tag{B-1}$$

it is easy that each term in the union bound (87) is given by $1/(\det(\mathbf{P}'))^r$. From the chain rule, the derivative of the term is

$$\frac{\partial}{\partial m_{ij}}\left(\frac{1}{\det^r(\mathbf{P}')}\right) = -\frac{r}{\det^{r+1}(\mathbf{P}')}\frac{\partial}{\partial m_{ij}}\det(\mathbf{P}'). \tag{B-2}$$

We need to only differentiate $\det(\mathbf{P}')$ with respect to each element m_{ij} of $\mathbf{M}$. We will first obtain the derivative of each term of $\mathbf{P}'$ with respect to m_{ij}. In particular, we will show that only elements along one column of $\mathbf{P}'$ have non-zero derivative. From (B-1), the $(k, l)^{\text{th}}$ element of $\mathbf{P}'$ is given by

$$p'_{kl} = \delta_{kl} + \frac{NS}{4K}\sum_{n=1}^{N} d'_{kn}d'^{*}_{ln}. \tag{B-3}$$

Note that $\mathbf{D}' = \mathrm{mat}(\mathbf{Md})$, and hence each element of $\mathbf{D}'$ is some linear combination of the elements of $\mathbf{M}$. Further, for complex variables m_{ij}, $\frac{\partial}{\partial m_{ij}} m_{ij} = 0$. Using this fact, the first term d'_{kn} can be treated as constant while differentiating with respect to m_{ij}, giving

$$\frac{\partial}{\partial m_{ij}} p'_{kl} = \frac{NS}{4K} \sum_{n=0}^{N-1} d'_{kn} \frac{\partial}{\partial m_{ij}} d'^{*}_{ln}. \tag{B-4}$$

Since $\mathbf{D}' = \mathrm{mat}(\mathbf{Md})$, d'_{ln} is the $(l + nN)^{\text{th}}$ term of $\mathbf{Md}$ (recall that indexing starts from 0). The key observation is that it is a linear combination of elements from $(l + nN)^{\text{th}}$ of $\mathbf{M}$. So, the derivative above is non-zero only if the row i to which m_{ij} belongs, is equal to $l + nN$, or $l = i \bmod N$, and $n = [i/N]$, where [.] is the standard integer floor function. Suppose i is indeed $l + nN$, then d'^{*}_{ln} is the i^{th} element of $\mathbf{Md}$. Using $\frac{\partial}{\partial m_{ij}} m^{*}_{ij} = 2$, it is easy to see that $\frac{\partial}{\partial m_{ij}} d'^{*}_{ln} = 2d_j^{*}$ in this case. Using all these things, (B-4) becomes

$$\frac{\partial}{\partial m_{ij}} p'_{kl} = \frac{NS}{4K} \sum_{n=0}^{N-1} d'_{kn} \delta_{(l = i \bmod N)} \delta_{n = [i/N]} 2d_j^{*}, \text{ or} \tag{B-5}$$

$$\frac{\partial}{\partial m_{ij}} p'_{kl} = \frac{NS}{2K} \, d'_{k,\,[i/N]} \delta_{l = (i \bmod N)} \, d_j^{*}. \tag{B-6}$$

In words, when differentiating $\mathbf{P}'$ with respect to m_{ij}, the term $\delta_{l = (i \bmod N)}$ above indicates that only the elements of column $(i \bmod N)$ have non-zero derivative. Along this column of $\mathbf{P}'$, the element in row k has derivative proportional to the element in the same row, but column $[i/N]$ of $\mathbf{D}'$, as indicated by the term $d'_{k,[i/N]}$.

Recall from (B-2) that we are interested in $\frac{\partial}{\partial m_{ij}}\det(\mathbf{P}')$. Now, since only one column of $\mathbf{P}'$ has a non-zero derivative, we can replace each element in that column by its derivative, and compute the determinant in order to get $\frac{\partial}{\partial m_{ij}}\det(\mathbf{P}')$. More precisely, we obtain the swapped matrix $\mathbf{P}_i'$ by replacing column $(i \bmod N)$ of $\mathbf{P}'$ by column $[i/N]$ of $\mathbf{D}'$ = mat($\mathbf{Md}$). Then,

$$\frac{\partial}{\partial m_{ij}}\det(\mathbf{P}') = \frac{NS}{2K}\det(\mathbf{P}_i')d_j^*. \tag{190}$$

Substituting this back in (B-2), we have obtained the derivative of each term in the union bound (87) with respect to m_{ij}.

As an aside, note that one also assemble the $N \times N$ matrix

$$\mathbf{Q}' = \mathbf{I}_N + \frac{NS}{4K}\mathbf{D}'^*\mathbf{D}'. \tag{B-7}$$

instead of the $t \times t$ matrix $\mathbf{P}'$ of (B-1). Clearly, $\det(\mathbf{P}') = \det(\mathbf{Q}')$, since both depend only on the singular values of $\mathbf{D}'$. However, when $N < t$, $\mathbf{Q}'$ is a smaller matrix and its determinant is easier to compute. In order to obtain the gradient, we are interested in

$$\frac{\partial}{\partial m_{ij}}\left(\frac{1}{\det^r(\mathbf{Q}')}\right) = -\frac{r}{\det^{r+1}(\mathbf{Q}')}\frac{\partial}{\partial m_{ij}}\det(\mathbf{Q}'). \tag{B-8}$$

Following the same analysis as the case, we now obtain the swapped matrix $\mathbf{Q}_i'$ by replacing row $[i/N]$ of $\mathbf{P}'$ by row $(i \bmod N)$ of $\mathbf{D}'$. Then,

$$\frac{\partial}{\partial m_{ij}}\det(\mathbf{Q}') = \frac{NS}{2K}\det(\mathbf{Q}_i')d_j^*, \tag{191}$$

which completes the necessary computation.

APPENDIX C

The Special Constraint Set for Length-2 Rate-t Codes

We describe the steps involved in deriving the constraint $\mathcal{N}(2t, 2t)$ for codes of length $N = 2$ and rate t. The encoding matrix $\mathbf{M}$ has dimension $2t \times 2t$. Consider the break-up of the encoding matrix into the $t \times t$ submatrices as follows.

$$\mathbf{M} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}. \tag{C-1}$$

From this point, the final structure (100) was obtained by making a couple of assumptions, neither of which has any theoretical basis, but seemed the right thing to do.

First, we assume that each of the $t \times t$ submatrices is unitary. Considering that the ultimate goal is equal spread of information in all the output symbols, this seemed a fair assumption. Note that the LCF encoding matrix for the same dimensions also has unitary submatrices. Now, any $2t \times 1$ difference vector $\mathbf{d}$ can be broken up as $\mathbf{d} = [\mathbf{d}_1{}^{\mathrm{T}}\ \mathbf{d}_2{}^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathbf{d}_1$ and $\mathbf{d}_2$ are both $t \times 1$ vectors. In this notation, it is easy to see that the difference matrix $\mathrm{mat}(\mathbf{Md})$ is given by

$$\mathrm{mat}(\mathbf{Md}) = \frac{1}{\sqrt{2}} [\mathbf{M}_{11}\mathbf{d}_1 + \mathbf{M}_{12}\mathbf{d}_2\ \mathbf{M}_{21}\mathbf{d}_1 + \mathbf{M}_{22}\mathbf{d}_2]. \tag{C-2}$$

Separating out the first submatrix $\mathbf{M}_{11}$ alone, this becomes

$$\text{mat}(\mathbf{Md}) = \frac{1}{\sqrt{2}}\mathbf{M}_{11}[\mathbf{d}_1 + \mathbf{M}_{11}{}^{*}\mathbf{M}_{12}\mathbf{d}_2 \;\; \mathbf{M}_{11}{}^{*}\mathbf{M}_{21}\mathbf{d}_1 + \mathbf{M}_{11}{}^{*}\mathbf{M}_{22}\mathbf{d}_2]. \qquad \text{(C-3)}$$

The union bound depends on the singular values of $\text{mat}(\mathbf{Md})$. Now, suppose a given set $\{\mathbf{M}_{ij}\}$ of submatrices achieves the optimum union bound. Consider the set of matrices $\{\mathbf{M}'_{ij}\}$, where each $\mathbf{M}'_{ij} = \mathbf{M}_{11}{}^{*}\mathbf{M}_{ij}$. Note that $\text{mat}(\mathbf{M}'\mathbf{d}) = \mathbf{M}_{11}{}^{*}\text{mat}(\mathbf{Md})$, hence $\text{mat}(\mathbf{M}'\mathbf{d})$ and $\text{mat}(\mathbf{Md})$ have the same singular values for all $\mathbf{d}$. Consequently, the submatrices $\{\mathbf{M}_{ij}\}$ and $\{\mathbf{M}'_{ij}\}$ have the same union bound. However, by construction, $\mathbf{M}'_{11} = \mathbf{M}_{11}{}^{*}\mathbf{M}_{11} = \mathbf{I}_t$. Thus, when the submatrices are unitary, the first one can be chosen to be the identity without loss of optimality. Choosing $\mathbf{M}_{11} = \mathbf{I}_t$, (C-3) becomes

$$\text{mat}(\mathbf{Md}) = \frac{1}{\sqrt{2}}[\mathbf{d}_1 + \mathbf{M}_{12}\mathbf{d}_2 \;\; \mathbf{M}_{21}\mathbf{d}_1 + \mathbf{M}_{22}\mathbf{d}_2]. \qquad \text{(C-4)}$$

Our second assumption is that the encoding matrix $\mathbf{M}$ itself is unitary. This assumption is roughly justified because optimum matrices anyway tend to have orthonormal columns (see Section 6.3.1). We require $\mathbf{M}^{*}\mathbf{M} = \mathbf{I}_{2t}$. Comparing this with the corresponding product based on (C-1), we that $\mathbf{M}^{*}\mathbf{M} = \mathbf{I}_{2t}$ if and only if $\mathbf{M}^{*}{}_{11}\mathbf{M}_{12} + \mathbf{M}^{*}{}_{21}\mathbf{M}_{22} = \mathbf{0}$. Substituting $\mathbf{M}_{11} = \mathbf{I}_t$, this implies $\mathbf{M}_{22} = -\mathbf{M}_{21}\mathbf{M}_{12}$. Substituting in (C-4), we get

$$\text{mat}(\mathbf{Md}) = \frac{1}{\sqrt{2}}[\mathbf{d}_1 + \mathbf{M}_{12}\mathbf{d}_2 \;\; \mathbf{M}_{21}(\mathbf{d}_1 - \mathbf{M}_{12}\mathbf{d}_2)]. \qquad \text{(C-5)}$$

At this point, there are just two $t \times t$ unitary matrices to choose, namely $\mathbf{M}_{12}$ and $\mathbf{M}_{21}$. We took up the case $t = 2$ and did a grid search to maximize the coding gain, when the symbols in $\mathbf{d}_1$ and $\mathbf{d}_2$ are drawn arbitrarily from a QAM difference alphabet. The search yielded the result that $\mathbf{M}_{12} = \exp(j\pi/4)\mathbf{I}_2$. Generalizing to arbitrary t, we get $\mathbf{M}_{12} = \exp(j\pi/4)\mathbf{I}_t$. Substituting $\mathbf{M}_{11} = \mathbf{I}_t$, $\mathbf{M}_{12} = \exp(j\pi/4)\mathbf{I}_t$, $\mathbf{M}_{22} = -\mathbf{M}_{21}\mathbf{M}_{12} = -\exp(j\pi/4)\mathbf{M}_{21}$, and denoting $\mathbf{M}_{21}$ by $\mathbf{Q}$, the assumed structure (C-1) becomes (100).

References

[1] G. L. Stuber, *Principles of Mobile Communication*, Second Edition, Kluwer Academic Publishers, 2001.

[2] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, 1995.

[3] V. Tarokh, N. Seshadri, A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Trans. on Info. Theory*, vol. 44, pp. 744-765, Mar. 1998.

[4] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. on Telecomm.*, vol. 10, pp. 585-95, Nov/Dec 1999.

[5] G. J. Foschini and M. J. Gans, "On the limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Comm.*, vol. 6, no. 3, pp. 311-355, Mar. 1998.

[6] E. Biglieri, J. Proakis, S. Shamai, "Fading Channels: Information-Theoretic and Communications Aspects," *IEEE Trans. on Info. Theory*, vol. 44, no. 6, pp. 2619 - 2692, Oct. 1998.

[7] S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.

[8] J.-C. Guey, M. P. Fitz, M. R. Bell, and W.-Y. Kuo, "Signal design for transmitter diversity wireless communication systems over Rayleigh fading channels," in *IEEE Trans. Comm.*, vol. 47, no. 4, pp. 527-537, Apr. 1999.

[9] S. M. Alamouti, "A simple transmitter diversity scheme for wireless communications," *IEEE Journal on Selected Areas in Comm.*, vol. 16, no.8, pp. 1451-1458, 1998.

[10] V. Tarokh, H. Jafarkhani and A. R. Calderbank, “Space-time block codes from orthogonal designs,” *IEEE Trans. on Info. Theory.*, vol. 45, no. 5, pp. 1456-1467, July 1999.

[11] G. Golden, G. Foschini, R. Venezuela, P. Wolniansky, “Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture,” *IEEE Electronics Letters*, vol. 35–1, pp. 14-16, Jan. 1999.

[12] B. Hassibi and B. M. Hochwald, “High-rate codes that are linear in space and time,” *IEEE Trans. on Info. Theory*, vol. 48, no. 7, July 2002.

[13] S. Sandhu and A. Paulraj, “Unified design of linear space-time block codes,” in *Proc. Globecom 2001*, vol. 2, pp. 1073-1077, November 2001.

[14] B. Papadias, G. Foschini, “On the capacity of certain space-time coding schemes,” *Eurasip Journal on App. Sig. Proc.*, vol. 2002:5, pp. 447-458, May 2002.

[15] A. Narula, M. Trott and G. Wornell, “Performance limits of coded diversity methods for transmitter antenna arrays,” *IEEE Trans. on Info. Theory*, vol. 45, no. 7, pp. 2418-2433, November 1999.

[16] S. Sandhu and A. Paulraj, “Space-time block coding: a capacity perspective,” *IEEE Comm. Letters*, vol. 4, no. 12, Dec 2000.

[17] R. W. Heath., Jr. and A. J. Paulraj, “Linear dispersion codes for MIMO systems based on frame theory,” *IEEE Trans. on Sig. Proc.*, vol. 50, no. 10, pp. 2429-2441, Oct. 2002.

[18] V. Tarokh, A. Naguib, N. Seshadri and A. Calderbank, “Combined array processing and space-time coding,” *IEEE Trans. on Info. Theory.*, vol. 44, no. 2, pp. 14-16, Jan. 1999.

[19] X. Ma and G. B. Giannakis, “Layered Space-Time Complex Field Coding: Full-Diversity with full rate, and trade-offs,” *Proc. of the 2nd Sensor Array and Multichannel SP Workshop*, Rosslyn, VA, pp. 442-446, August 2002.

[20] X. Ma and G. B. Giannakis, "Full-Diversity Full-Rate Complex-Field Space-Time Coding," *IEEE Trans. on Sig. Proc.*, vol. 51, no. 11, pp. 2917-2930, Nov 2003.

[21] M. O. Damen, A. Tewfik and J. C. Belfiore, "A construction of a space-time code based on number theory," *IEEE Trans. on Info. Theory*, vol. 48, no. 3, pp. 753-760, March 2002.

[22] H. El Gamal and M. O. Damen, "Universal space-time coding," *IEEE Trans. on Info. Theory*, vol. 49, no. 5, pp. 1097-1119, May 2003.

[23] J. R. Barry, E. A. Lee, D. G. Messerschmitt, *Digital Communication*, Third Edition, Kluwer Academic Publishers, 2003.

[24] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications," *ACM SIGSAM*, vol. 15, pp. 37-44, 1981.

[25] M. O. Damen, A. Chkief and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Letters*, pp. 161-163, May 2000.

[26] E. Viterbo and E. Biglieri, "A universal lattice decoder," in *GRETSI* 14-eme Colloque, Juan-les-Pins, France, September 1993.

[27] E. Viterbo and J.Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Information Theory*, vol. 45, pp. 1639-1642, July 1999.

[28] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mat. Computation*, vol. 44, pp. 463-471, Apr. 1985

[29] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181-191, 1994.

[30] M. O. Damen, H. El Gamal and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. on Info. Theory*, vol. 49, no. 10, pp. 2389-2402, Oct 2003.

[31] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon Limit Error-Correcting Coding and Decoding,” *IEEE International Conference on Commun.*, pp. 1064-1070, May 1993.

[32] J. Hagenauer, E. Offer and L. Papke, “Iterative Decoding of Binary Block and Convolutional Codes,” *IEEE Trans. on Info. Theory*, vol. 42, no. 2, pp. 429-445, March 1996.

[33] B. M. Hochwald and S. T. Brink, “Achieving Near-Capacity on a multiple-antenna channel,” *IEEE Trans. on Comm.*, vol. 51, no. 3, pp. 389-399, March 2003.

[34] B. Hassibi and B. M. Hochwald, “How much training is needed in multiple-antenna wireless links?,” *IEEE Trans. on Info. Theory*, vol. 49, no. 4, pp. 951-963, April 2003.

[35] B. Hassibi and B. M. Hochwald, “Cayley differential unitary space-time codes,” *IEEE Trans. on Info. Theory*, vol. 48, no. 6, p. 1485-1503, June 2002.

[36] L. Zheng and D. Tse, “Diversity and Multiplexing: A Fundamental trade-off in Multiple Antenna Channels,” *IEEE Trans. on Info. Theory*, vol. 49, no. 5, pp. 1073-1096, May 2003.

[37] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 1993.

[38] H.-J Su and E. Geraniotis, “Space-time turbo codes with full antenna diversity,” *IEEE Trans. on Comm.*, vol. 49, no. 1, pp. 47-57, Jan 2001.

[39] G. Ungerboeck, “Channel Coding with Multilevel/Phase Signals,” *IEEE Trans. on. Info. Theory* IT-28, No. 1, Jan 1982.

[40] A. R. Hammons Jr. and H. El Gamal, “On the theory of space-time codes for PSK modulation,” *IEEE Trans. on Info. Theory*, vol. 46, no. 2, pp. 524-542, March 2000.

[41] S. Sandhu, A. Paulraj and K. Pandit, “On non-linear space-time block codes,” *IEEE Int. Conf. on Acoustics, Speech, and Sig. Proc. ICASSP 2002*, vol. 3, pp. 2417-2420, May 2002.

[42] B. Varadarajan and J. R. Barry, "Optimization of full rate full diversity linear space-Time Codes using the Union Bound," *Proc. IEEE Info. Theory Workshop*, pp. 210-213, March 31-April 4, 2003.

[43] A. Edelman, *Eigenvalues and Condition Numbers of Random Matrices*, Ph. D. thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[44] K. L. Chung, *A Course in Probability Theory*, Third Edition, Academic Press, 2001.

[45] G. D. Forney Jr., "Coset codes. I. Introduction and Geometrical Classification," *IEEE Trans. Info. Theory*, vol. 34, no. 5, pp. 1123-1151, Sep 1988.

[46] G. D. Forney Jr., "Coset codes. II. Binary Lattices and Related Codes," *IEEE Trans. Info. Theory*, vol. 34, no. 5, pp. 1152-1187, Sep 1988.

[47] R. G. Gallagher, "Low Density Parity Check Codes," *IRE Trans. Info. Theory*, vol. IT-8, pp. 21-28, Jan 1962.

[48] D. J. MacKay, "Good Error Correcting Codes Based on Very Sparse Matrices," *IEEE Trans. Info. Theory*, p. 399-431, May 1999.

[49] V. Gulati and K. R. Narayanan, "Concatenated space-time codes for quasi-static fading channels: constrained capacity and code design," *IEEE Global Telecomm. Conf. GLOBECOMM '02*, vol. 2, pp. 1202-1206, Nov. 2002.

[50] S. ten Brink and B. M. Hochwald, "Detection thresholds of iterative MIMO processing," *IEEE Int. Symposium on Info. Theory ISIT 2002*, pp. 22,

[51] Z. Wang and G. B. Giannakis, "Outage Mutual Information for MIMO Space-Time Channels," *Proc. of 40th Allerton Conference*, pp. 885-894, October 2-4, 2002.

[52] B. Varadarajan and J. R. Barry, "The Rate-Diversity Trade-Off for Linear Space-Time Codes," in *Proc. IEEE Veh. Tech. Conf.*, vol. 1, pp. 67-71, September 2002.

[53] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd Edition, McGraw-Hill, 1991.

[54] M. Godavarti and A. O. Hero, “Diversity and degrees of freedom in wireless communications,” *2002 IEEE Int. Conf. on Acoustics, Speech and Sig. Proc.*, vol. 3, pp. 2861-2864, 2002.

[55] T. M. Cover and J. A. Thomas, *Information Theory*, Wiley and Sons, 1991.

[56] G. Caire, G. Taricco, E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Int. Conf. on Comm.* ICC ‘97, vol. 3, pp. 1463-1467, June 1997.

[57] B. Hassibi and H. Vikalo, “On the expected complexity of sphere decoding,” *Signals, Systems and Computers*, 2001, vol. 2, pp. 1051-1055, 2001.

[58] H. Vikalo and B. Hassibi, “The expected complexity of sphere decoding, Part I: Theory, Part II: Applications,” *IEEE Trans. Signal Processing*, submitted for publication.

[59] J. B. Anderson and S. Mohan, “Sequential Coding Algorithms: A Survey and Cost Analysis,” *IEEE Trans. on. Comm.*, vol. 32, no. 2, pp. 169-176, Feb 1984.

[60] J. Hagenauer, “The Revival of Sequential Decoding,” *Proc.* 5^{th} *Int. ITG Conf. on Source and Channel Coding*.

[61] D. W. Waters and J. R. Barry, “Noise-Predictive Decision-Feedback Detection for Multiple-Input Multiple-Output Channels,” *IEEE Trans. Signal Processing*, submitted July 2003.

[62] A. Wiesel, A. Pagès, J. R. Fonollosa, “Efficient Implementation of Sphere Demodulation,” IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC 2003, June 15-18, 2003, Rome, Italy.

[63] A. Y. Grama, V. Kumar, “Parallel Algorithms for Discrete Optimization Problems,” *Technical Report* TR 93-11, Univ. of Minn., 1993.

[64] A. M. Chan and I. Lee, “A New Reduced-Complexity Sphere Decoder For Multiple Antenna Systems,” *IEEE Int. Conf. on Comm 2002*, vol. 1, pp. 460-464, May 2002.

[65] M. C. Davey, “Error-Correction using Low-Density Parity-Check Codes,” Ph. D. thesis, Cambridge University, December 1999.

[66] S. ten Brink and B. M. Hochwald, “Detection thresholds of iterative MIMO processing,” *IEEE Int. Symposium on Info. Theory ISIT 2002*, pp. 22.

[67] N. Prasad and M. Varanasi, “Analysis of Decision Feedback Detection for MIMO Rayleigh Fading Channels and Optimum Allocation of Transmitter Powers and QAM Constellations,” *39th Annual Allerton Conf. on Comm. Control, and Comput.*, Monticello, IL, Oct. 2001.

[68] N. Prasad and M. Varanasi, “Optimum Efficiently Decodable Layered Space-Time Block Codes,” *Proc. Asilomar Conf. on Signals, Systems and Computers*, Monterey, CA, Nov. 2001.

[69] B. Varadarajan and J. R. Barry, “Linear Space-Time Codes Optimized For Successive Cancellation Decoding,” *Int. Zurich Seminar on Comm.*, vol. 1, pp. 132-135, Zurich, Switzerland, Feb 2004.

VITA

Badri Varadarajan is from Chennai, India. In August 1999, he received his Bachelor's degree (B. Tech.) in Electrical Engineering from the Indian Institute of Technology, Chennai (IITM). In December 2000, he was awarded the Master's degree in Electrical and Computer Engineering by the Georgia Institute of Technology, Atlanta. In July 2004, he received the Ph. D. degree, also from Georgia Tech.

During his Ph. D., he worked as a research assistant under the supervision of his advisor Dr. John Barry, with funding from the Yamacraw Foundation and the National Science Foundation. In the summer of 2003, he was a summer intern with Mayflower Communications at Bedford, MA. In August 2004, he will join Texas Instruments, Dallas, as an electrical design engineer.

Like most other people, Badri Varadarajan is primarily interested in finding never-ending success, pleasure and glory. In his more philosophical moments, he tries to take the Buddha's middle path to Nirvana. Aside from these main interests, he also dabbles in coding and information theory, with specific application to wireless communication.