ABSTRACT

Title of Dissertation: EFFICIENT CODING AND DECODING METHODS FOR MULTI-ANTENNA WIRELESS COMMUNICATION SYSTEMS

Zoltan Safar, Doctor of Philosophy, 2009

Dissertation directed by: Professor K. J. Ray Liu
Department of Electrical and Computer Engineering

As existing wireless communication systems and standards cannot fully support the new, emerging multimedia applications, the designers of future wireless systems will have to face the challenge of devising coding and modulation methods that can provide reliable communication at very high data rates. To achieve this goal, one of the most promising techniques suggested recently is the exploitation of the spatial dimension: by employing multiple transmit and receive antennas, the detrimental effects of channel fading can be significantly reduced.

This thesis aims to contribute to the field of wireless communications by developing efficient coding and decoding methods for systems having multiple transmit

and receive antennas. First, we propose a systematic space-time trellis code construction method for the quasi-static, flat multi-antenna channel model by exploiting the trellis structure. The method can be used to construct space-time trellis codes for an arbitrary number of transmit antennas and any memoryless modulation. Then, we consider the problem of space-time code design for correlated fading channels, and derive the performance criteria assuming that the space-time correlation matrix is of full rank. We also propose a new code design criterion, the uniqueness criterion, and develop a systematic space-time trellis code design method for the correlated fading channel model. Finally, we propose a fast decoding algorithm for space-frequency-coded multi-antenna OFDM systems. The central component of the algorithm is a modulation independent decoding framework, which is based on a new interpretation of the sphere decoding problem. Using this decoding framework, we devise several SF block code decoding algorithms: a modulation independent decoding algorithm that can be used with any memoryless modulation, and 2 modulation specific decoding algorithms for QAM and PSK constellations using the idea of fast nearest neighbor search.

EFFICIENT CODING AND DECODING METHODS FOR MULTI-ANTENNA WIRELESS COMMUNICATION SYSTEMS

by

Zoltan Safar

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:

Professor K. J. Ray Liu, Chairman / Advisor
Associate Professor Steven A. Tretter
Assistant Professor Haralabos Papadopoulos
Assistant Professor Sennur Ulukus
Professor Carlos Berenstein

DEDICATION

To my parents, Sáfár Judit and Sáfár Zsolt

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my advisor, Prof. K. J. Ray Liu. During our interactions, I have been influenced by his vision, attitude, energy and desire to excel, and he has shown me how to expect high quality work from myself and how to achieve excellence through hard work. I especially appreciate the freedom he gave me to choose a research topic within the broad areas of his research interests. I also value his commitment to his students: when I was seeking advice, he was always available. He has played a significant role in my professional and personal development during my Ph.D. studies at the University of Maryland, and his help and guidance have been instrumental to the work presented in this thesis.

I am also indebted to my technical secondary school teacher, Horváthné Tőkei Zsuzsa. She has been a role model for me, and her influence steered me toward choosing electrical engineering as a career. I believe that her excellent classes and teaching methods have planted the seeds of the desire to become a teacher and educator in me.

Moreover, I would like to thank the members of my research group, the DSP (or CSPL?) group, for their help and company throughout the years of my stay here. I feel lucky to have had a chance to become a part of this excellent group of young researchers, and I have experienced a real sense of community among them. I would like to explicitly mention Wade Trappe, Yan Sun and Hong Zhao: thank you guys for the times spent together in our office. It has been a pleasure to work

together with Dr. Weifeng Su. I think that we complement each other very well, and I hope that our collaboration will continue after my departure.

Finally, I am grateful to my parents for their love and unconditional support. They have made countless sacrifices to make my and my brother's lives better and to make sure that we can explore and develop our talents according to our desires. As a consequence, all of my accomplishments are also their accomplishments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

In recent years, wireless communication has experienced a rapid growth, and it promises to become a globally important infrastructure. The advances in integrated circuit technology and digital signal processing algorithms have made wireless communication technology accessible to millions of people. The light weight, longer operational time, and affordable prices of portable devices have resulted in ever increasing demand for wireless services.

We are living in the information era: acquiring and exchanging information has become an integral part of our daily lives. Now it is possible to retrieve and communicate information in the forms of voice, data, and multimedia content (e.g. images, video, and music). However, the current wireless communication systems and standards, the 3G cellular standard or the Wireless LAN IEEE802.11 standard, cannot cope with the requirements of the new emerging multimedia applications. The quality of service they provide is not competitive with that wire-line service providers can offer now. The large volume and sensitivity of multimedia data

require the development and deployment of wireless communication systems that can guarantee reliable data transmission at very high data rates.

As a consequence, the designers of future wireless communication systems will have to face the challenge of devising coding, modulation and signal processing techniques that can combat the adverse effects of the radio signal propagation environment, such as multi-path fading and interference more effectively. One of the most promising techniques suggested recently is the exploitation of the spatial dimension: by employing multiple transmit and receive antennas and developing appropriate coding and modulation methods, the detrimental effects of channel fading can be significantly reduced. It was established that the capacity of multi-antenna communication systems increases roughly linearly with the number of antennas. This result triggered enormous research activity in the area of multiple-input-multiple- output (MIMO) wireless communication systems.

One approach to improving the performance of the MIMO communication system is by using *space-time* (ST) codes. In case of ST coding, the transmitter sends two-dimensional codewords, with the dimensions corresponding to different transmit antennas and different discrete time instants. Such approach can increase the slope of the error performance curve, resulting in a much more energy-efficient and/or bandwidth-efficient system than the traditional single-antenna communication systems. However, most ST coding techniques have been developed for idealistic channel models, such as frequency non-selective (flat), quasi-static or fast fading channels, assuming that there is no correlation between the different transmit and receive antenna pairs. Since real-world MIMO channels exhibit both spatial and temporal correlation, the coding methods designed for ideal channel models may suffer considerable performance degradation. As a consequence, it is

of interest to develop analytical tools to be able to evaluate the effect of fading correlation on the performance and to devise coding and modulation methods that explicitly take the channel correlation into account.

In case of broadband wireless communication systems, the channel exhibits frequency selectivity (delay spread), resulting in inter-symbol interference (ISI) that can also degrade the performance seriously. Among the various ISI mitigating approaches, Orthogonal Frequency Division Multiplexing (OFDM) is one of the most promising techniques, as it eliminates the need for high complexity equalization and offers high spectral efficiency. OFDM has been successfully used for broadband applications and has been chosen as the standard for digital audio broadcasting and digital terrestrial TV broadcasting in Europe. Due to parallel transmission over multiple sub-carriers, OFDM is able to operate at higher data rates and this fact has motivated the IEEE 802.11 working group to choose OFDM as the physical layer implementation for the IEEE802.11b WLAN standard.

In order to combine the advantages of both the MIMO systems and the OFDM, *space-frequency* (SF) coded MIMO-OFDM systems have been proposed, where two-dimensional coding is applied to distribute channel symbols across space (transmit antennas) and frequency (OFDM tones). However, the constraints of practical implementation may hamper the applicability of the previously proposed SF codes. So far fast decoding algorithms have only been proposed for ST codes transmitted over flat, quasi-static fading channels, so the decoding complexity of SF codes can be a bottleneck in case of a portable wireless device with limited computing power and/or battery life. The complexity of the maximum likelihood (ML) decoding algorithm is exponential in the data rate, so the development of computationally efficient decoding algorithms is of paramount importance.

1.2 The Wireless Channel

The mobile radio channel severely attenuates and distorts the transmitted signal. The components of the detrimental effects of the wireless channel can be classified as [1], [2], [3], [4]:

- *Path loss*: Signal attenuation due to the distance between the transmitter and the receiver. The transmitter-receiver separation might even be several kilometers.

- *Shadowing*: Signal attenuation due to environment clutter such as buildings and trees. It causes slow signal variations as the mobile moves distances of 10-100 meters.

- *Fading*: The transmitter and the receiver are surrounded by objects which reflect and scatter the transmitted radio signal. As a consequence, the received signal consists of incoming radio waves arriving from different directions with different amplitudes and phases. These signal components add constructively or destructively, causing fast variations in the received signal strength as the mobile moves distances of 1-10 centimeters. If there is a direct path between the transmitter and the receiver, the situation is called line-of-sight (LOS) propagation, while the scenario when direct wave from the transmitter to the receiver is blocked is called non-line-of-sight (NLOS) propagation.

If the bandwidth of transmitted signal is small, the arrival time differences of the incoming reflected signals are small compared to the duration of one channel symbol period. All reflections arrive essentially at the same time, and all frequencies within the transmission bandwidth are affected in the same way. In this case, the channel is called *frequency non-selective* (flat) channel. On the other hand, if

the transmission bandwidth is high, the arrival time differences of the incoming radio waves may be significant compared to the duration of one channel symbol period. In this case, the received signal will consist of a sequence of impulses, each impulse collecting the reflected waves with small arrival time differences, i.e signals corresponding to approximately the same propagation path length. As a consequence, the received signal will spread in time, sometimes over multiple channel symbol periods, and the frequency response of the channel will vary over the bandwidth of the transmitted signal. This case corresponds to a *frequency selective* channel, whose complex baseband impulse response at time t can be modeled as [2], [5]

$$h(t,\tau) = \sum_{p=0}^{P-1} \alpha_p(t)\delta(\tau - \tau_p(t)), \tag{1.1}$$

where P is the number of resolvable impulses (also called as the number of paths, even though the signals contributing to each impulse may come through different physical propagation paths), $\alpha_p(t)$ is the complex amplitude of the p-th impulse (also called the p-th path), and $\tau_p(t)$ is the p-th delay. Each impulse consists of the superposition of many incoming signals whose arrival time differences are relatively small, so the magnitudes of the impulses change unpredictably as the mobile moves due to the continuously changing phase differences, causing signal fading. In this work, we assume NLOS propagation, which corresponds to the "worst case" scenario, and in this case, the values of $\{\alpha_p(t)\}$, for each t, can be modeled as zero mean, complex Gaussian random variables [4]. Since the magnitude of each impulse (or path gain) $\alpha_p(t)$ follows Rayleigh distribution, the adopted fading channel model is also called Rayleigh fading model. The frequency flat fading model can be derived from (1.1) by assuming only one impulse, i.e. $P = 1$. In a multi-antenna wireless propagation environment, there is a wireless channel corresponding to each

transmit and receive antenna pair.

In the sequel, we will assume that during the transmission of one frame, there are no variations in the effects of path loss and shadowing, so they will be assumed constant, and only the effect of the (possibly time varying) fading will be considered. Sometimes it will be assumed that the MIMO channel is quasi-static, which means that it stays constant over the duration one frame period.

1.3 MIMO Capacity

Information theoretic works [8], [9], [10], [11], [12] have promised enormous capacity increase when using multiple transmit and receive antennas. It was shown that the capacity of a multiple antenna system grows at least linearly with the number of transmit antennas, provided that the number of receive antennas is greater than or equal to the number of transmit antennas [8], [9].

The BLAST architecture [13], [14], [15], [16], [17], [18], [19] has been proposed to achieve or approach the capacity available in MIMO quasi-static, flat fading channels. The basic idea of the method is to take several independent data streams as inputs and distribute them over the transmit antennas via spatio-temporal multiplexing. The data streams may be encoded using some traditional (one-dimensional) error correcting code, but coding across the transmit antennas is not applied. The V-BLAST system [14], [16] simply transmits separate data streams through different transmit antennas, while the D-BLAST architecture [13], [18], [19] creates diagonal layers from the data streams. At the receiver side, the interference caused by the simultaneous transmission of multiple data streams need to be resolved by a sequence of nulling and canceling steps. The performance (bit error rate) of such such systems is usually comparable to the

performance of a single-antenna system, but the data rate the BLAST systems can offer is considerably higher. Recently many BLAST-like systems have been proposed by combining BLAST with adaptive modulation [20], per antenna rate and power control [21], and turbo coding [22], [23].

1.4 Space-Time Coding

The ST coding approach looks at the communication problem from a different perspective. Instead of attempting to maximize the data rate, the objective is to improve the performance compared to a single-antenna communication system. This objective is achieved by developing two-dimensional coding methods, where the dimensions correspond to different transmit antennas and different time instants. Since there is only one input data stream, instead of attempting to cancel the interference caused by simultaneous transmission from different transmit antennas, the goal of the ST code construction methods is to *design* the interference in such a way that the transmitted symbol sequence could be recovered at the receiver side as perfectly as possible.

As it is well known, the bit error rate (BER) of a communication system operated over additive white Gaussian noise (AWGN) channels decreases super-exponentially with the signal-to-noise ratio (SNR). However, in case of a single-antenna Rayleigh fading channel, the BER is only inversely proportional to the average SNR [4], [5], which can be expressed as

$$BER \approx \frac{1}{c}\frac{1}{\rho},$$

with c being a positive real constant, and ρ denoting the average SNR. This means that the performance of the communication system degrades considerably com-

pared to the AWGN case.

This situation can be improved significantly by providing the receiver with some form of *diversity*. If multiple replicas of the transmitted signal are sent to the receiver through independent virtual channels, the probability of simultaneous fading will be much smaller. The virtual channels may correspod to transmissions through the same physical channel at different times (time diversity) or different frequency bands (frequency diversity). Note, however, that if the virtual channels are highly correlated, the diversity techniques may not result in sigificant performance improvement.

A form of time diversity, channel coding combined with interleaving, adds redundancy with a certain algebraic structure that can be exploited to detect and correct transmission errors [6]. Spatial diversity corresponds to adding redundancy in the spatial domain by building a system with multiple transmit and/or receive antennas. In case of ST coding, the diversity is achieved by coding at the transmitter side, so it is also referred to as transmit diversity. The BER of a ST-coded communication system transmitting over Rayleigh fading channels can be approximated as [59]

$$BER \approx \frac{1}{c} \frac{1}{\rho^D},$$

where D, the diversity advantage, describes the asymptotic slope of the BER curve as the function of the average SNR (in log-log representation), and c, the coding advantage, describes the vertical shift of the BER curve. In the sequel, these quantities will be used to characterize and compare the performance of different ST codes. In case of the quasi-static channel model, the maximum value of D (the maximum achievable diversity order) is the product of the number of transmit antennas and the number of receive antennas [59].

The ST coding methods also assume that the receiver knows the MIMO channel perfectly. If this information is not available, ST modulation approaches [24], [25], [26], [27] can be used.

1.4.1 Space-Time Block Coding

The ST block encoder is a memoryless mapping that maps a set of source symbols to two-dimensional codewords, which are transmitted through different transmit antennas at different time instants. An important class of ST block codes are constructed from orthogonal designs. The theory of orthogonal designs, which focuses on the construction of square matrices from real or complex variables in such a way that their columns are orthogonal to each other, has a long history in mathematics [28], [29], [30], [31]. The first transmit diversity scheme using orthogonal designs was proposed in [32]. The author constructed space-time block codes for 2 transmit antennas from the 2 by 2 complex orthogonal design. The idea was extended and further developed in [33], [34], [34], [36], [37]. The importance of this ST construction method lies in the facts that when transmitted over quasi-static, flat MIMO channels, these codes can achieve full diversity, and they can also be decoded in a computationally efficient way by making independent decisions on the constituting source symbols.

However, it was established that complex orthogonal designs with full symbol rate and full diversity only exist for 2 transmit antennas [33], so ST codes constructed form block-orthogonal (or quasi-orthogonal) designs were proposed [38], [39], [40], [41], [42] to further increase the transmission rate. Some of these ST block codes could also achieve full diversity and full symbol rate, but at the price of increased decoding complexity.

1.4.2 Space-Time Trellis Coding

In case of ST trellis codes, the encoder consists of a finite state machine and a memoryless signal mapping. At each discrete time instant, the encoder takes the current input source symbol and the current encoder state, and based on this information it moves to the next state and outputs a channel symbol for each transmit antenna. At the receiver side, the Viterbi algorithm [6], [7] can be used to decode the transmitted source symbol sequence. The authors of [59] presented code design rules for 2 transmit antennas, and later works [60], [61], [63], [64] provided generalizations or codes with improved performance. A variable rate trellis code construction method was proposed in [65] for fast fading channels combining the idea of multiple trellis coded modulation with repetition coding. Since one of the main topics of this dissertation is ST trellis code design, the reader can find a more detailed description of previuos and related work in Chapters 2 and 3.

The choice between ST block and trellis codes depends largely on the application and the parameters of the implemented MIMO system. However, in general, they can be compared as follows:

- The ST trellis codes have better performance than ST block codes for the same spectral efficiency (the ST block codes have lower coding advantage).
- The ST block codes have higher decoding complexity. In fact, the minimum number of encoder states in a full-diversity ST trellis code is exponential in the number of transmit antennas [33].

1.5 Space-Frequency Coding

The OFDM modulation (also called as Multicarrier Modulation [43] or Discrete Multitone [44]) has been adopted by many existing communication systems due to its advantages. By using cyclic prefix to make the transmitted time-domain signal periodic, the OFDM modulation transforms the frequency selective wideband channel to a set of frequency flat fading sub-channels. In [45], it was shown that the same architecture can be easily generalized to MIMO channels, resulting in MIMO-OFDM systems.

The objective of the *space-frequency* (SF) code design is to exploit the diversity available in frequency-selective MIMO channels. In addition to spatial diversity, the variations of the channel in the frequency domain can also be exploited, resulting in frequency diversity. In case of SF-coded MIMO-OFDM systems, two-dimensional coding is applied to encode the source symbols over space (transmit antennas) and frequency (OFDM tones). The first SF coding scheme was proposed in [46], in which previously existing ST codes were used by replacing the time domain with frequency domain. Later works [47]–[50] also described similar schemes, i.e., using ST codes directly as SF codes. The resulting SF codes could achieve only spatial diversity and were not guaranteed to achieve the full (spatial and frequency) diversity available in the MIMO frequency selective fading channels.

The performance criteria for SF coded MIMO-OFDM systems were derived in [51]. The ultimate limits on the maximum achievable diversity order were also established. The authors showed that, in general, existing ST codes cannot exploit the frequency diversity available in the frequency selective MIMO channels, and it was suggested that a completely new code design procedure will have to be developed for MIMO-OFDM systems. Characterizing the performance of MIMO-

OFDM systems was also considered in [53], and the maximum achievable diversity order was found to be the same as in [51]. Later in [52], the authors provided a construction method for a class of full-diversity SF codes by multiplying a part of the DFT-matrix with the input symbol vectors. A full-diversity SF code construction method was also proposed in [56], [57]. The SF codes were constructed from ST codes via a simple repetition mapping. As a consequence, the authors provided a method to use any previously constructed ST code for full-diversity transmission over frequency selective MIMO channels.

The idea of coding across multiple OFDM blocks, resulting in *space-time-frequency* (STF) codes, was first proposed in [54] for two transmit antennas and further developed in [55] for multiple transmit antennas. The method described in [55] can provide full diversity only if the number of encoded OFDM blocks was not smaller than the number of transmit antennas.

1.6 Thesis Overview and Contributions

The goal of this dissertation is to develop efficient coding and decoding methods for MIMO wireless communication systems. We achieve this goal by devising ST trellis code design methods and proposing a fast SF block code decoding algorithm. In case of ST trellis codes, the efficiency is measured by the energy efficiency of the resulting codes for a given spectral efficiency and decoding complexity. Moreover, scalability and flexibility are also of importance: the emphasis will be put on systematic approaches that can provide solutions for many possible system parameters, such as the number of transmit antennas and the applied modulation method. In case of the SF code decoding method, the efficiency is measured by the required computational complexity for a giver BER perfromance.

In Chapter 2, we adopt the quasi-static flat fading channel model and propose a systematic design method by exploiting the structure of the state transitions in the trellis. The method can be used to design ST codes that provide full diversity advantage for an arbitrary number of transmit antennas, arbitrary number of encoder states and arbitrary memoryless modulations. The design rules for full diversity advantage do not specify the ST codes completely, offering the possibility to further optimize for coding advantage. Moreover, we develop a code design procedure that benefits from this possibility for a certain class of encoders. Based on the design rules for diversity advantage, we reduce the code construction problem to a combinatorial optimization problem and propose a computationally efficient suboptimal solution. To our knowledge, this is the first work that considers systematic code design for both diversity advantage and coding advantage.

In Chapter 3, we consider the problem of ST trellis code design for correlated flat fading channels, taking into account both spatial and temporal channel correlation. We focus our attention on the development of systematic ST code design methods that are flexible and scalable. First, we derive the performance criteria for a channel model in which the channel changes from channel symbol period to channel symbol period in a correlated manner, assuming that the space-time correlation matrix is of full rank. We show that for this transmission scenario, the effect of the channel correlation and the ST code on the performance can be separated, resulting in channel-independent performance criteria. We characterize the performance of the ST codes by finding exact expressions for the achieved diversity advantage and coding advantage. Then, we propose a new design criterion and analyze the properties of the ST trellis codes satisfying the proposed criterion. Finally, we develop a systematic code construction procedure that jointly considers diversity

advantage and coding advantage for an arbitrary number of transmit antennas and any memoryless modulation. The proposed design method has extremely low complexity: we provide a closed form solution to the code design problem. To the best of our knowledge, our method is the first systematic space-time trellis code design method proposed for non-quasi-static channel models.

In Chapter 4, we propose a computationally efficient decoding algorithm for SF block codes. We formulate the sphere decoding problem in the complex domain, which allows us to fully exploit the distance structure of complex signal constellations. We develop a systematic method to transform the decoding problem into an equivalent representation that is more appropriate for the purpose of sphere decoding. Then, we propose a modulation independent sphere decoding framework by interpreting the sphere decoding problem as a greedy, constrained depth-first search. Due to the modular structure of the framework, it can be used to construct a decoding algorithm that can be used with any memoryless modulation, and it can also be tailored to a particular modulation method by taking full advantage of the geometric properties of the chosen signal constellation. Finally, for square QAM and PSK constellations, we propose a fast, nearest neighbor search algorithm that can considerably reduce the decoding complexity.

Finally, we draw conclusions and discuss some possible future research directions in Chapter 5.

Chapter 2

Systematic Space-Time Trellis Code Design for Quasi-Static MIMO Channels

2.1 Introduction

The quasi-static, flat fading channel model was the first MIMO channel model proposed in the literature [58], [59], and it has gained considerable popularity since then. The model assumes that the channel is frequency flat (no delay spread) and that the channel stays constant over one frame period. In general, the spatial correlation between different transmit-receive antenna pairs is also neglected, so in some sense, the quasi-static, flat model can be regarded as the simplest MIMO channel model. The performance criteria for this channel model were derived in [58] and [59], characterizing the ST codes with two quantities: the diversity advantage, which describes the asymptotic error rate decrease as a function of the signal to noise ratio (SNR), and the coding advantage, which determines the vertical shift

of the error performance curve. In [59], the authors proposed design rules for two transmit antennas to achieve the maximum diversity advantage. They also derived a lower bound on the complexity of the encoder and the decoder for the desired diversity advantage and data throughput. This lower bound states that in order to achieve a diversity advantage of K and transmit one B-ary source symbol per state transition, the encoder and the decoder must have at least $N_{min} = B^{K-1}$ states. The repetition coded delay diversity scheme described in [60] was the first systematic design rule for an arbitrary number of transmit antennas. Using this method, ST codes achieving full diversity advantage can be designed for arbitrary constellations and encoders with N_{min} states. This work also introduced the idea of zero symmetry to constrain computer search for ST codes with more than two antennas. In [61], the design problem was transformed into the binary domain. The code design was based on the finite field counterpart of the ST code performance criteria for full spatial diversity. The authors proposed code design procedures for an arbitrary number of transmit antennas and an arbitrary number of states, but only for BPSK and QPSK constellations. Moreover, the design methods in [60] and [61] for full diversity advantage uniquely determine the ST codes, not leaving room to improve the coding advantage.

In this chapter, we propose a systematic design method based on an alternative approach: we exploit the structure of the trellis to design ST codes that provide full diversity advantage for an arbitrary number of transmit antennas, arbitrary number of encoder states (as long as it satisfies the lower bound) and arbitrary memoryless modulation. Our method can be treated as a generalization of the results of [59] and [60]. The design rules for full diversity advantage do not specify the ST codes completely, offering the possibility to further optimize for coding

advantage.

Moreover, we develop a code design procedure that benefits from this possibility for the important special case of encoders with N_{min} states. Based on the design rules for diversity advantage, we reduce the code construction problem to a combinatorial optimization problem and propose a computationally efficient suboptimal solution.

The chapter is organized as follows. Section 2.2 will introduce the mathematical model of the communication system. The performance criteria for ST trellis codes will also be described in this section. The code construction method will be developed in Sections 2.3 and 2.4. Section 2.5 will describe specific ST code construction examples, and the simulation results demonstrating the performance of these codes will be provided in Section 2.6. Finally, we will summarize the chapter in Section 2.7.

2.2 System Model and Performance Criteria

In this section, the mathematical model of the wireless communication system under study will be described. The used notation will also be introduced. Then, we will briefly restate the performance criteria derived in [58] and [59]. These criteria serve as a basis for the development of our systematic design procedure.

Consider a wireless communication system with K transmit and L receive antennas (the transmit antennas are indexed by k, $k \in \{0, 1, ..., K-1\}$, and the receive antennas are indexed by l, $l \in \{0, 1, ..., L-1\}$). The input bit stream is divided into b_s bit long blocks, forming B-ary ($B = 2^{b_s}$) source symbols. The ST encoder works as a finite state machine with N states: it takes the current b_s bit long source symbol, b_t ($b_t \in \{0, 1, ..., B-1\}$) at discrete time t ($t = 0, 1, 2, 3, ...$),

and governed by this input and the current state, S_t ($S_t \in \{0, 1, ..., N-1\}$), it moves to the next state, S_{t+1}. During this state transition, the encoder outputs K B-ary channel symbol indices, one for each transmit antenna. We denote by $i^k(S_t, b_t)$ the channel symbol index for transmit antenna k, generated during the state transition from S_t when the current input source symbol is b_t. We will also use the channel symbol index vector, defined as

$$\mathbf{i}(S_t, b_t) = [\, i^0(S_t, b_t), i^1(S_t, b_t), ..., i^{K-1}(S_t, b_t)\,]^T.$$

These channel symbol indices select one of the B different waveforms for each antenna, and the selected waveforms are transmitted simultaneously through the transmit antennas. In the sequel, $\Omega(i)$ will be the complex baseband vector-space representation of the ith passband waveform ($i \in \{0, 1, ..., B-1\}$). $\Omega(i)$ will also be referred to as the ith constellation point or channel symbol. All the constellations are assumed to be normalized so that the average energy of the constellation is unity (if the channel symbols are equally likely). $\Omega(i^k(S_t, b_t))$ will denote the constellation point output by antenna k when the current state is S_t and the current input is b_t. The vector of channel symbols is given by:

$$\mathbf{c}(S_t, b_t) = [\, \Omega(i^0(S_t, b_t)), \Omega(i^1(S_t, b_t)), ..., \Omega(i^{K-1}(S_t, b_t))\,]^T.$$

The transmission medium is assumed to be flat (frequency non-selective), quasi-static, Rayleigh fading channel, and α_{kl} will represent the path gain from transmit antenna k to receive antenna l. These path gains are modeled as independent, complex, zero mean, circularly symmetric Gaussian random variables with unit variance. Furthermore, some additional assumptions are made to facilitate the analysis. First, the receiver has knowledge of the α_{kl} propagation coefficients. Second, the receiver is perfectly synchronized with the transmitter.

Based on the above assumptions, after down-conversion, matched filtering and sampling, the received signal at receive antenna l, at discrete time t can be expressed as [59]

$$r_t^l = \sum_{k=0}^{K-1} \sqrt{\frac{\rho}{K}} \alpha_{kl} \Omega(i^k(S_t, b_t)) + z_t^l = s_t^l + z_t^l. \tag{2.1}$$

In (2.1), s_t^l and z_t^l stand for the received signal and noise components, respectively, and z_t^l's are independent, complex, zero mean, circularly symmetric Gaussian random variables with unit variance. Consequently, the average signal to noise ratio per source symbol at receive antenna l becomes:

$$SNR_l = \frac{E[|s_t^l|^2]}{E[|z_t^l|^2]} = \rho. \tag{2.2}$$

Assume that the previously described transmitter sends T $(T > K)$ B-ary source symbols to the receiver. The ST encoder, while encoding the data, goes through the following sequence of states:

$$S_0 \xrightarrow{b_0} S_1 \xrightarrow{b_1} S_2 \xrightarrow{b_2} \ldots \xrightarrow{b_{T-2}} S_{T-1} \xrightarrow{b_{T-1}} S_T.$$

In words, the encoder starts in S_0, takes the first input b_s-tuple, b_0, moves to S_1, and so on. As a result of this state transition sequence, the encoder produces the channel symbol vector sequence:

$$\mathbf{c}(S_0, b_0), \mathbf{c}(S_1, b_1), \ldots, \mathbf{c}(S_{T-1}, b_{T-1}).$$

The above vector sequence can be arranged into a K by T matrix, $\mathbf{C}$:

$$\mathbf{C} = [\, \mathbf{c}(S_0, b_0), \mathbf{c}(S_1, b_1), \ldots, \mathbf{c}(S_{T-1}, b_{T-1}) \,].$$

The decoder, due to decoding errors, goes through a different sequence of states,

$$\hat{S}_0 \xrightarrow{\hat{b}_0} \hat{S}_1 \xrightarrow{\hat{b}_1} \hat{S}_2 \xrightarrow{\hat{b}_2} \ldots \xrightarrow{\hat{b}_{T-2}} \hat{S}_{T-1} \xrightarrow{\hat{b}_{T-1}} \hat{S}_T,$$

producing the erroneously decoded source symbol sequence $\{\hat{b}_t\}$ and the K by T channel symbol matrix $\hat{\mathbf{C}}$:

$$\hat{\mathbf{C}} = [\, \mathbf{c}(\hat{S}_0, \hat{b}_0), \mathbf{c}(\hat{S}_1, \hat{b}_1), \ldots, \mathbf{c}(\hat{S}_{T-1}, \hat{b}_{T-1}) \,].$$

We can define $\mathbf{D}$, the channel symbol difference matrix as $\mathbf{D} = \mathbf{C} - \hat{\mathbf{C}}$, and a K by K matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{D}\mathbf{D}^{\mathcal{H}}$. Let $\mathbf{A}$ (and $\mathbf{D}$) be of rank m. Since $\mathbf{A}$ is Hermitian and nonnegative definite, its eigenvalues are real and nonnegative. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ be the nonzero eigenvalues of $\mathbf{A}$. Given the earlier described channel model, it can be shown [59] that the probability that the decoder erroneously decodes $\hat{\mathbf{C}}$ if $\mathbf{C}$ was sent can be upper bounded as:

$$P(\hat{\mathbf{C}}|\mathbf{C}) \leq \left(\prod_{i=1}^{m} \lambda_i \right)^{-L} \left(\frac{E_0}{4KN_0} \right)^{-mL}. \tag{2.3}$$

The performance criteria [58],[59] were derived to minimize $P(\hat{\mathbf{C}}|\mathbf{C})$ for a given SNR:

1. *Design for full spatial diversity (rank criterion)*: The matrix $\mathbf{D}$ must be of full row rank for any distinct $\mathbf{C}$ and $\hat{\mathbf{C}}$ matrices. (Then we have $m = K$.) In this case, a diversity advantage of KL has been achieved.

2. *Design for coding advantage (determinant criterion)*: The minimum determinant of $\mathbf{A}$ taken over all distinct $\mathbf{C}$ and $\hat{\mathbf{C}}$ matrices must be as large as possible. If the minimum determinant is γ, then a coding advantage of $\sqrt[K]{\gamma}$ has been achieved.

2.3 Design for Diversity Advantage

2.3.1 Trellis Structure Analysis

The goal of this subsection is to analyze the algebraic structure of the trellis of the ST encoder and find closed form expressions that relate the state sequence $\{S_t\}$ to the starting state S_0 and the input source symbol sequence $\{b_t\}$. The additions and the multiplications are assumed to be standard integer operations. The modulo operation will always be written explicitly to avoid ambiguity.

Assume that the encoder has $N = RB^{K+p-1}$ states, with $R = 2^r$, $B = 2^{b_s}$, $b_s > 0$, $p \geq 0$ and $0 \leq r < b_s$. Therefore, it satisfies the lower bound of [59] for desired diversity advantage of K with B-ary source symbols. Any large enough power of 2 number can be put into this form; the purpose of this representation is to make the analytical treatment easier. The number N is simply decomposed into the product of two numbers: the first number, R is less than B, and the other number is a power of B.

The state transition of the encoder at time t is determined by the previous state, S_{t-1} ($S_{t-1} \in \{0, 1, \ldots, N-1\}$), and the previous B-ary input, b_{t-1} ($b_{t-1} \in \{0, 1, \ldots, B-1\}$). Analytically it can be described as:

$$S_t = (BS_{t-1} + b_{t-1}) \bmod N = B\left(S_{t-1} \bmod (RB^{K+p-2})\right) + b_{t-1}. \tag{2.4}$$

It is shown in Appendix A that we can unfold this recursion and obtain a closed form expression for S_t, $1 \leq t \leq K+p-1$:

$$S_t = B^t\left(S_0 \bmod (RB^{K+p-t-1})\right) + \sum_{m=0}^{t-1} B^{t-1-m} b_m, \tag{2.5}$$

and for S_{K+p}:

$$S_{K+p} = B^{K+p-1}(b_0 \bmod R) + \sum_{m=1}^{K+p-1} B^{K+p-1-m} b_m. \tag{2.6}$$

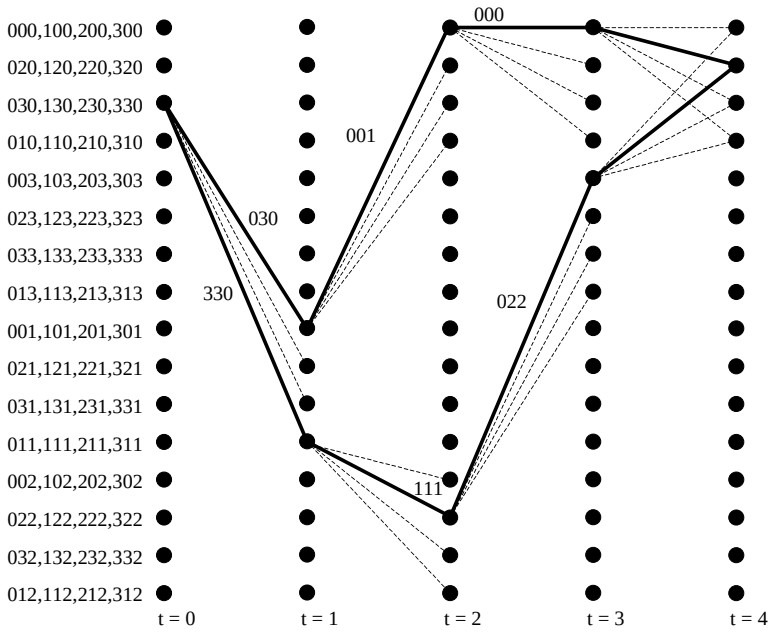

Figure 2.1: Example ST code for 3 antennas, QPSK

Based on these analytical results, we can deduce some important information about the error path structure of the trellis. Assume that the first decoding error occurs at state S_0, i.e. the correct and the erroneous paths diverge at S_0, and they merge at some later state. As a consequence of this assumption, we have $S_0 = \hat{S}_0$ and $b_0 \neq \hat{b}_0$. We have no information regarding the rest of the b_t's and $\hat{b}_t$'s. In this case, (2.5) immediately tells us that for $1 \leq t \leq K + p - 1$, $S_t \neq \hat{S}_t$. Moreover, from (2.6), it can be seen that S_{K+p} may or may not be equal to $\hat{S}_{K+p}$, since $b_0 \bmod R$ may be equal to $\hat{b}_0 \bmod R$, even though $b_0 \neq \hat{b}_0$. Thus, we have the following theorem:

Theorem 2.1: If $R = 1$ (i.e. $r = 0$), the shortest error path is exactly $K + p$ long. If $R > 1$ (i.e. $r > 0$), the shortest error path is either $K + p$ long or longer. For arbitrary $p \geq 0$ and $0 \leq r < b_s$, the shortest error path is at least K long, i.e.

the paths diverging at S_0 can merge only at S_K or later.

2.3.2 Design for Full Diversity

Using formulas (2.5) and (2.6), we will derive design rules that guarantee that the ST trellis code achieves full diversity advantage. First, we will obtain sufficient conditions to make the channel symbol difference matrix corresponding to the first K long error path segment of the first error event full rank. Afterwards, the results will be extended to arbitrary channel symbol difference matrices.

In the ST encoder, B channel symbol index vectors are assigned to each state, according to the branches emanating from that state. The current source symbol selects one of them, and the kth ($k = 0, 1, ..., K-1$) index of the chosen vector determines the constellation point for antenna k. Figure 2.1 depicts an example ST code for 3 antennas and QPSK constellation ($K = 3, B = 4, N = 16$). In this case, if the current state is state 2 and the value of the current source symbol is 3, the ST encoder selects the 3rd channel symbol index vector, $[3, 3, 0]^T$, and moves to state 11. The 0th, 1st and 2nd antennas will transmit the channel symbols corresponding to the indices 3, 3 and 0, respectively.

Suppose that the transmitter sends T ($T > K$) source symbols. Without loss of generality, we can assume that the first decoding error event occurs at S_0, making the correct and the decoded paths diverge. For now, we are concerned only about the first K long segment of all error paths of length K or longer, immediately after the first error event has occured. Our goal is to construct the K by K channel symbol difference matrix $\mathbf{D_1}$, defined as

$$\mathbf{D_1} = [\mathbf{c}(S_0, b_0) - \mathbf{c}(\hat{S}_0, \hat{b}_0), \mathbf{c}(S_1, b_1) - \mathbf{c}(\hat{S}_1, \hat{b}_1), \ldots, \mathbf{c}(S_{K-1}, b_{K-1}) - \mathbf{c}(\hat{S}_{K-1}, \hat{b}_{K-1})], \tag{2.7}$$

in such a way that it is of full rank for any possible correct and erroneous paths through the trellis. Our method is to make $\mathbf{D_1}$ upper triangular with nonzero diagonal elements. We exclude all ST codes that do not produce upper triangular $\mathbf{D_1}$ matrices, so the resulting ST codes may not be optimal. However, what we gain is a problem formulation that leads to a simple solution.

The $S_0 \rightarrow S_1$ state transition is special since both the correct and the erroneous paths start at the same state. The goal is to set the 0th entry of the 0th column of $\mathbf{D_1}$ to a nonzero value and to zero out the rest of the entries in that column. This can be achieved by the following conditions that form the first half of the design rules:

(1a) The 0th indices of the channel symbol index vectors at the same state must be different.

(1b) The remaining indices of the channel symbol index vectors at the same state must be the same.

In our example, assume that the $b_0 = 0$ (top) path is the correct path and the $\hat{b}_0 = 3$ (bottom) path is the erroneously decoded path. The channel symbol index vectors $[0, 3, 0]^T$ and $[3, 3, 0]^T$ have different 0th indices, but the 1st and 2nd indices are the same; therefore, the 0th column of the $\mathbf{D_1}$ matrix will be $[1 + j, 0, 0]^T$.

For the rest of the state transitions $S_t \rightarrow S_{t+1}$, $t = 1, 2, ..., K - 1$, the objective is to set the tth entry of the tth column of $\mathbf{D_1}$ to a nonzero value and to zero out all the entries below the tth entry in that column. To facilitate the explanation, we introduce the following definitions:

Definition 2.1 : A *level t group* is a collection of all destination states that can be reached at state transition t from a given S_0 starting state through all possible $b_0, b_1, \ldots, b_{t-1}$ input sequences.

Definition 2.2 : A *subgroup of a level t group* is a collection of all destination states that can be reached at state transition t from a given S_0 starting state and a given b_0 starting branch through all possible $b_1, b_2, \ldots, b_{t-1}$ input sequences.

In order to effectively use these definitions in the design procedure, we need to describe the relationship between the encoder states and the groups and subgroups at different levels. Equation (2.5) expresses the state transition at time t, $t = 1, 2, \ldots, K-1$, as a function of the starting state, S_0, and the source symbol sequence $b_0, b_1, \ldots, b_{t-1}$. Because S_0 is kept constant in Definition 2.1 for all possible $b_0, b_1, \ldots, b_{t-1}$ sequences, we can eliminate the effect of the starting state by taking modulo B^t of both sides of (2.5). Therefore, the expression $S_t \bmod B^t$ will describe how S_t depends on the $b_0, b_1, \ldots, b_{t-1}$ sequence for an arbitrary, but fixed, S_0 starting state. From (2.5), we obtain:

$$S_t \bmod B^t = B^{t-1}b_0 + B^{t-2}b_1 + \ldots + Bb_{t-2} + b_{t-1}. \tag{2.8}$$

The above quantity can be thought of as a t digit B-ary number. As the input B-tuples $(b_0, b_1, \ldots, b_{t-1})$ vary from $(0, 0, \ldots, 0)$ to $(B-1, B-1, \ldots, B-1)$, the value of $S_t \bmod B^t$ varies from 0 to $B^t - 1$. Consequently, for $t = 1, 2, \ldots, K-1$, any level t group starts at state m such that $m \bmod B^t = 0$ and consists of B^t consecutive states.

Similarly, S_0 and b_0 are kept constant in Definition 2.2, so the expression $S_t \bmod B^{t-1}$ will describe how S_t changes as a function of $b_1, b_2, \ldots, b_{t-1}$. From (2.5), we have:

$$S_t \bmod B^{t-1} = B^{t-2}b_1 + B^{t-3}b_2 + \ldots + Bb_{t-2} + b_{t-1}. \tag{2.9}$$

The above quantity can be thought of as a $t-1$ digit B-ary number. As the input B-tuples $(b_1, b_2, \ldots, b_{t-1})$ vary from $(0, 0, \ldots, 0)$ to $(B-1, B-1, \ldots, B-1)$, the value of $S_t \bmod B^{t-1}$ varies from 0 to $B^{t-1} - 1$. Therefore, we conclude that for

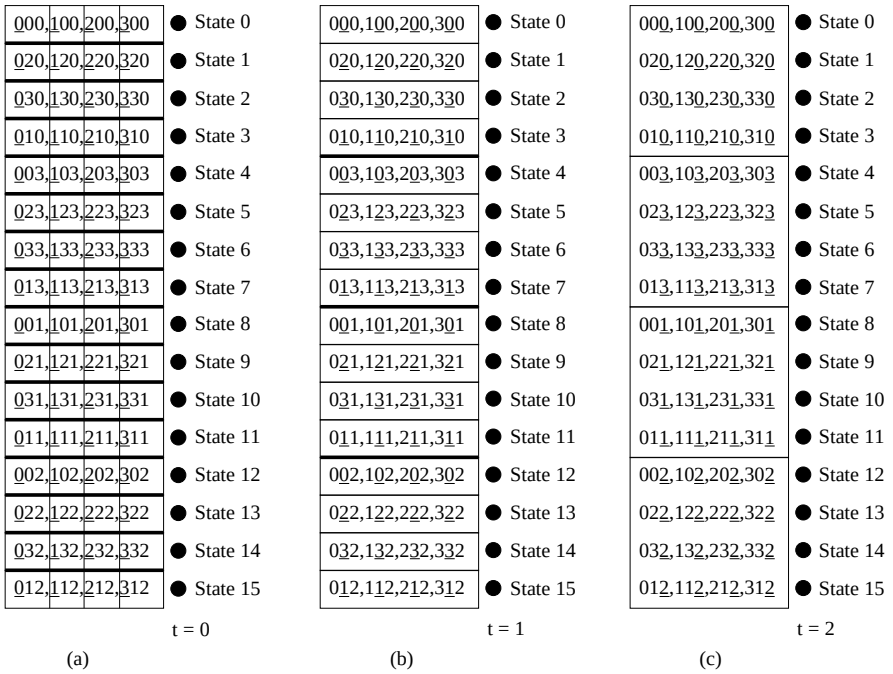

Figure 2.2: The group/subgroup structure of the example ST code

$t = 1, 2, \ldots, K-1$, any subgroup of a level t group starts at state m such that $m \bmod B^{t-1} = 0$ and consists of B^{t-1} consecutive states.

Since $b_0 \in \{0, 1, \ldots, B-1\}$, by definition every group consists of B subgroups according to different b_0 values. From (2.8), it can be seen that different b_0 values result in disjoint sets of $S_t \bmod B^t$ values as the B-tuples $(b_1, b_2, \ldots, b_{t-1})$ vary from $(0, 0, \ldots, 0)$ to $(B-1, B-1, \ldots, B-1)$. Thus, for $t = 1, 2, \ldots, K-1$, every level t group consists of B disjoint subgroups. We index the subgroups within a group by the 0th source symbol, so S_t belongs to the b_0th subgroup and $\hat{S}_t$ belongs to the $\hat{b}_0$th subgroup of the same level t group.

In the case of the ST code of Figure 1, the level 1 groups consist of 4 consecutive states, starting at states $0, 4, 8$ and 12. The subgroups consist of only one state. The only level 2 group is comprised of all the 16 states, and its subgroups are made

up of 4 consecutive states, starting at states $0, 4, 8$ and 12. The level 1 and level 2 groups and subgroups of the example ST code are depicted in Figure 2.2 (b) and (c), respectively.

Because both the correct and the erroneous paths start from the same state ($S_0 = \hat{S}_0$), at state transition t, $t = 1, 2, \ldots, K-1$, both the correct path $\{S_t\}$ and the erroneous path $\{\hat{S}_t\}$ go through states that belong to the same level t group. This means that if the mth indices of the channel symbol index vectors at states belonging to any level t group are the same, then the mth entry of the tth column of $\mathbf{D_1}$ will be zero. For example, in Figure 1, states 8 and 11 belong to the same level 1 group, and the 2nd indices of the channel symbol index vectors $[0, 0, 1]^T$ and $[1, 1, 1]^T$ are the same. As a consequence, the 1st column of the $\mathbf{D_1}$ matrix becomes $[1-j, 1-j, 0]^T$.

Since the first decoding error occurs at S_0 ($b_0 \neq \hat{b}_0$), at state transition t, $t = 1, 2, \ldots, K-1$, the correct path $\{S_t\}$ and the erroneous path $\{\hat{S}_t\}$ go through states that belong to different subgroups of the same level t group. We can take advantage of this fact as follows: if the mth indices of the channel symbol index vectors at states belonging to different subgroups of the same level t group are different, then the mth entry of the tth column of $\mathbf{D_1}$ will be nonzero. To continue the example, states 0 and 13 belong to different subgroups of the same (only) level 2 group. The 2nd indices of the channel symbol index vectors $[0, 0, 0]^T$ and $[0, 2, 2]^T$ are different, so the 2nd column of the matrix $\mathbf{D_1}$ will be $[0, 2, 2]^T$.

Having produced the methods to place zero and non-zero entries into the matrix $\mathbf{D_1}$, we can state the second half of the design rules:

(2a) For $t = 1, 2, \ldots, K-1$, the tth indices of the channel symbol index vectors at states belonging to the same subgroup of any level t group must be the

same, and they must be different from the tth indices of the channel symbol index vectors at states belonging to any other subgroup of that group.

(2b) For $t = 1, 2, \ldots, K-2$, the $(t+1)$st, $(t+2)$nd,..., $(K-1)$st indices of the channel symbol index vectors at states belonging to the same level t group must be the same. (Note that criterion (2b) is omitted for $t = K-1$.)

After making the matrix $\mathbf{D_1}$ full rank, the final task is to show that the channel symbol difference matrix $\mathbf{D}$ corresponding to the transmission of all T source symbols is also of full rank. The matrix $\mathbf{D}$ can be decomposed as:

$$\mathbf{D} = [\, \mathbf{D_1}, \mathbf{D_2} \,], \tag{2.10}$$

where $\mathbf{D_1}$ is defined in (2.7), and $\mathbf{D_2}$ is a K by $(T-K)$ matrix. Since $\mathbf{D_2}$ is arbitrary, this description includes the cases when the correct and the decoded paths diverge and merge several times. From linear algebra, it is well known that if $\mathbf{D_1}$ is of full rank, then $\mathbf{D}$ is also of full (row) rank. Consequently, the design rules will produce codes that provide full diversity advantage.

Figure 2.2 (a) illustrates design rules (1a) and (1b) for the ST code shown in Figure 2.1. At each sate, the 0th indices of the channel symbol index vectors are different, and the 1st and 2nd indices are the same. Figure 2.2 (b) shows how the rules (2a) and (2b) are applied for $t = 1$. In each subgroup of the level 1 groups, the 1st indices of the channel symbol index vectors are the same, and they are different from the 1st indices of the index vectors of any other subgroup of that group. Moreover, the 2nd indices of the channel symbol index vectors in each level 1 group are the same. Finally, rule (2a) for $t = 2$ is illustrated in Figure 2.2 (c). In each subgroup of the only level 2 group, the 2nd indices of the channel symbol index vectors are the same, and they are different from the 2nd indices of the index

vectors of the other subgroups.

By observing the group/subgroup structure of the state transitions in the trellis, we can make the design rules independent of the state evolution in time. The above design method describes relationships between channel symbol indices of different antennas at different states. Furthermore, these design rules do not fully determine the state - channel symbol assignment, providing the possibility to further optimize for coding gain.

Design rules (1a) and (1b) are similar to the design rules described in [59] for two transmit antennas. Therefore, our approach can be treated as a generalization of the method of [59] to an arbitrary number of transmit antennas.

2.4 Design for Coding Advantage

In general, finding the best way to assign channel symbol indices to antennas and states is not a simple task. If $N > N_{min}$, the shortest error path is longer than K, so the corresponding code difference matrix does not have any special structure. As a consequence, expressing the minimum determinant of the code becomes very difficult. However, in the $N = N_{min} = B^{K-1}$ case (i.e. $r = 0$ and $p = 0$), it is possible to find an efficient method to maximize the coding gain, so from now on, it is assumed that the encoder has N_{min} states.

The channel symbol difference matrix corresponding to the first K long segment of the error paths after the first decoding error has occured is the matrix $\mathbf{D_1}$, defined in (2.7). It is square and upper triangular, so its determinant is the product of its diagonal elements:

$$\det(\mathbf{D_1}) = \prod_{k=0}^{K-1} \left(\Omega(i^k(S_k, b_k)) - \Omega(i^k(\hat{S}_k, \hat{b}_k)) \right). \tag{2.11}$$

Let us define the K by K matrix $\mathbf{A_1}$ as: $\mathbf{A_1} = \mathbf{D_1}\mathbf{D_1}^{\mathcal{H}}$. Then γ_1, the determinant of $\mathbf{A_1}$, is

$$\gamma_1 = \det(\mathbf{A_1}) = \det(\mathbf{D_1})\det(\mathbf{D_1})^* = \prod_{k=0}^{K-1} \left|\Omega(i^k(S_k, b_k)) - \Omega(i^k(\hat{S}_k, \hat{b}_k))\right|^2. \quad (2.12)$$

Considering the transmission of all T source symbols, and using the decomposition of (2.10), the matrix $\mathbf{A} = \mathbf{D}\mathbf{D}^{\mathcal{H}}$, whose minimum determinant is to be maximized, can be expressed as

$$\mathbf{A} = \mathbf{D_1}\mathbf{D_1}^{\mathcal{H}} + \mathbf{D_2}\mathbf{D_2}^{\mathcal{H}} = \mathbf{A_1} + \mathbf{A_2},$$

where $\mathbf{A_2} = \mathbf{D_2}\mathbf{D_2}^{\mathcal{H}}$. By construction, both $\mathbf{A_1}$ and $\mathbf{A_2}$ are Hermitian and nonnegative definite. To continue the argument, we will use the following theorem from linear algebra [62]:

> Let $\mathbf{X}$ and $\mathbf{Y}$ be K by K, Hermitian and nonnegative definite matrices. Moreover, let $\lambda_0(\mathbf{X}) \geq \lambda_1(\mathbf{X}) \geq \ldots \geq \lambda_{K-1}(\mathbf{X})$ denote the real and nonnegative eigenvalues of $\mathbf{X}$. Then we have the following inequality for $i = 0, 1, \ldots, K-1$:

$$\lambda_i(\mathbf{X} + \mathbf{Y}) \geq \lambda_i(\mathbf{X}) + \lambda_{K-1}(\mathbf{Y}). \quad (2.13)$$

In our case, (2.13) becomes:

$$\lambda_i(\mathbf{A_1} + \mathbf{A_2}) \geq \lambda_i(\mathbf{A_1}) + \lambda_{K-1}(\mathbf{A_2}) \qquad \text{for} \qquad i = 0, 1, \ldots, K-1.$$

Since $\mathbf{A_2}$ is nonnegative definite, $\lambda_{K-1}(\mathbf{A_2}) \geq 0$. This means that

$$\lambda_i(\mathbf{A_1} + \mathbf{A_2}) \geq \lambda_i(\mathbf{A_1}) \qquad \text{for} \qquad i = 0, 1, \ldots, K-1. \quad (2.14)$$

From this, we can conclude that γ, the determinant of $\mathbf{A}$, satisfies the inequality:

$$\gamma = \det(\mathbf{A}) = \prod_{i=0}^{K-1} \lambda_i(\mathbf{A_1} + \mathbf{A_2}) \geq \prod_{i=0}^{K-1} \lambda_i(\mathbf{A_1}) = \det(\mathbf{A_1}) = \gamma_1. \quad (2.15)$$

We can fix an arbitrary correct path and pick an arbitrary error path that is longer than K state transitions. Both this error path and the error path corresponding to the K long error event that starts from the same S_0 starting state and the same $\hat{b}_0$ starting branch go through states that belong to the same subgroups of the same groups, resulting in $\mathbf{D_1}$ matrices with the same diagonal elements (design rules (1a) and (2a)). Therefore, for any error event that is longer than K state transitions, it is possible to find a K long error event with the same $\det(\mathbf{A_1})$ value. As a consequence of this observation and (2.15), γ_{min}, the minimum determinant of the code, can be determined by taking into account only the shortest error events:

$$\gamma_{min} = \min_{\substack{\{S_l,b_l\},\{S'_l,b'_l\} \\ l=0,1,\ldots,K-1}} \prod_{k=0}^{K-1} \left|\Omega(i^k(S_k,b_k)) - \Omega(i^k(\hat{S}_k,\hat{b}_k))\right|^2. \tag{2.16}$$

The minimum is taken over all possible K long correct and incorrect paths.

The S_k and $\hat{S}_k$ state transition sequences can also be described by making use of the group/subgroup structure of the trellis. The results of Section 2.3 allow us to map the first K long segment of the correct and erroneous paths of the first decoding error event onto different groups and subgroups of states. Toward this end, we introduce a channel symbol index based notation that does not explicitly depend on the state transition sequence.

Let i_l^0, $i_l^0 \in \{0,1,\ldots,B-1\}$, be the 0th indices of the channel symbol index vectors at the same state corresponding to source symbol l ($l \in \{0,1,\ldots,B-1\}$). For simplicity, it is assumed that the 0th indices of the channel symbol index vectors at different states corresponding to the same source symbol values are the same. Moreover, let i_l^k, $k = 1,2,\ldots,K-1$, $i_l^k \in \{0,1,\ldots,B-1\}$, denote the kth indices of the channel symbol index vectors at the states belonging to the lth subgroup of the same level k group ($l \in \{0,1,\ldots,B-1\}$). According to design

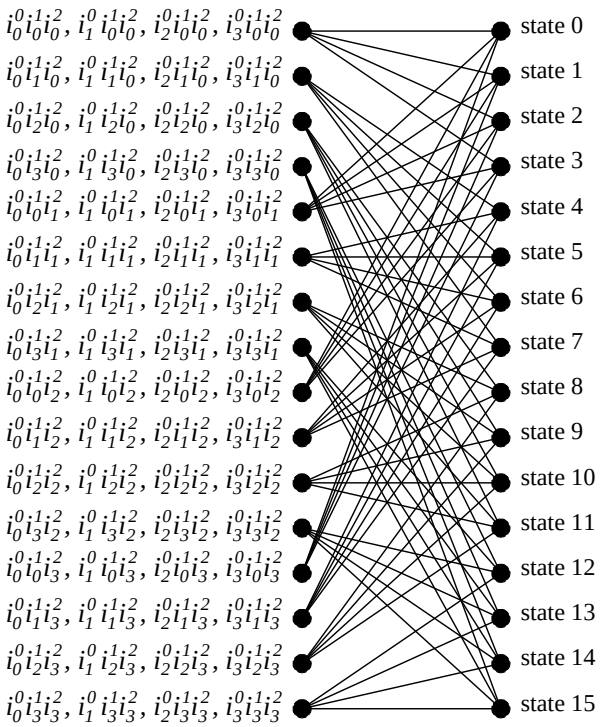

Figure 2.3: Example ST code template for 3 antennas, 4-ary constellations

rules (1a) and (2a), the relation $i_l^k \neq i_m^k$ must hold for any $l \neq m$. Therefore, the B-tuple $(i_0^k, i_1^k, \ldots, i_{B-1}^k)$, $k = 0, 1, \ldots, K-1$, is an (arbitrary) permutation of the numbers $0, 1, \ldots, B-1$.

Applying the design method of Section 2.3 and using the above index notation, we can create a "template" ST code. It is called template because the design rules for full spatial diversity do not specify the codes completely. It contains channel symbol index templates at each state for each antenna. For the ST code example of Figure 2.1, this template is shown in Figure 2.3. Here, the 4-tuples $(i_0^0, i_1^0, i_2^0, i_3^0)$, $(i_0^1, i_1^1, i_2^1, i_3^1)$ and $(i_0^2, i_1^2, i_2^2, i_3^2)$ can be any permutations of the numbers (0,1,2,3).

The ST code will achieve full diversity advantage for arbitrary permutations. The objective is to find those permutations that result in maximizing the coding advantage of the ST code.

Using the simplified notation $l = b_0$ and $m = \hat{b}_0$ ($l \neq m$), it was shown earlier that for $k = 1, 2, \ldots, K-1$, S_k belongs to the lth subgroup and $\hat{S}_k$ belongs to the mth subgroup of the same level k group. Therefore, using the above defined index notation, we can make the following substitutions:

$$i^k(S_k, b_k) = i_l^k \qquad \text{and} \qquad i^k(\hat{S}_k, \hat{b}_k) = i_m^k, \qquad \text{for} \qquad k = 0, 1, \ldots, K-1. \tag{2.17}$$

Consequently, the expression for the minimum determinant can be rewritten as:

$$\gamma_{min} = \min_{\substack{l,m \in \{0,1,\ldots,B-1\} \\ l<m}} \prod_{k=0}^{K-1} \left| \Omega(i_l^k) - \Omega(i_m^k) \right|^2. \tag{2.18}$$

The $l < m$ condition can be used since the squared distance function is symmetric in its arguments. The final goal is to maximize the minimum determinant. Therefore, if Π_B denotes the set of all permutations of the numbers $0, 1, \ldots, B-1$, and $\sigma^k \in \Pi_B$ ($k = 0, 1, \ldots, K-1$) stands for a particular permutation $(i_0^k, i_1^k, \ldots, i_{B-1}^k)$, then γ_{min}^*, the optimal minimum determinant, can be expressed as:

$$\gamma_{min}^* = \max_{\sigma^0, \sigma^1, \ldots, \sigma^{K-1}} \left[\min_{\substack{l,m \in \{0,1,\ldots,B-1\} \\ l<m}} \prod_{k=0}^{K-1} \left| \Omega(i_l^k) - \Omega(i_m^k) \right|^2 \right]. \tag{2.19}$$

This combinatorial optimization problem can be interpreted as follows. The design rules for diversity advantage and K permutations of the numbers $0, 1, \ldots, B-1$ together uniquely determine the code. The task is to find those permutations that offer the largest minimum determinant. Because the numbers $0, 1, \ldots, B-1$ can be arranged in $B!$ different ways, the size of the search space is $(B!)^K$. This means that exhaustive search may be impractical in certain cases. For example, if $B = 16$ and $K = 5$, the search has to be done over approximately $4 \cdot 10^{66}$ possibilities.

value	binary vector representation	polynomial representation	power representation	power
0	00	0	0	-
1	01	1	α^0	0
2	10	α	α^1	1
3	11	$\alpha+1$	α^2	2

Table 2.1: Element representation in GF(4)

To get around this complexity growth, we propose a suboptimal approach that offers a practical solution. The basic idea is to restrict the search space such that the resulting complexity is not prohibitive. Toward this end, we define the notion of the parametric permutation function.

The parametric permutation function is a function that generates a subset of all possible permutations of the numbers $0, 1, \ldots, B-1$. Different parameters produce different permutations, so the problem will be reduced to a search for the best parameter. Therefore, the parametric permutation function is required to have the following properties:

- It must be a bijective map: for any parameter, it must map the set $\{0, 1, \ldots, B-1\}$ onto itself in a one-to-one manner.
- Two different parameters must generate two different permutations.

In the sequel, we will use the notation $i_2 = \psi_B(n, i_1)$ for one possible realization of the parametric permutation function. The value n is the parameter to be optimized ($n \in \{1, 2, \ldots, B-1\}$), and i_1 and i_2 are the input and output indices, respectively ($i_1, i_2 \in \{0, 1, \ldots, B-1\}$). Note that the obvious $\psi_B(n, i_1) = (ni_1) \bmod B$, for odd n, is not a good choice because it does not "shuffle" the indices well enough.

The output index is generated according to the following description. The input index i_1 and the parameter n are treated as binary vector representations of two

i_1	$n=1$	$n=2$	$n=3$
0	0	0	0
1	1	2	3
2	2	3	1
3	3	1	2

Table 2.2: The generated permutations

field elements in GF(B)=GF(2^{b_s}). These field elements are multiplied together according to field arithmetic, and the output index i_2 will be the binary vector representation of the product. For example, for $B = 4$, the field GF(4) can be built up using α, a root of the primitive polynomial $p(x) = x^2 + x + 1$, as shown in Table 2.1. In this case, the function $\psi_4(\cdot, \cdot)$ will generate the permutations given in Table 2.2. The table entries are the function values for different input index and parameter values.

The above example shows two general properties of the function $\psi_B(\cdot, \cdot)$. First, the zero index always maps to itself: $\psi_B(n, 0) = 0$. Second, if the parameter is one, the function value is equal to the input index value: $\psi_B(1, i_1) = i_1$. Since the above definition of the parametric permutation function exploits the algebraic properties of the underlying Galois field, it can be easily seen that it has the required properties. Moreover, the elements of a Galois field can also be represented as powers of a primitive element, providing the possibility of efficient implementation by turning the multiplications into modulo $B - 1$ additions.

Replacing the permutation operation by the parametric permutation function, the optimization problem reduces to

$$\gamma^*_{min} = \max_{\substack{n_0,n_1,\ldots,n_{K-1} \\ \in\{1,2,\ldots,B-1\}}} \left[\min_{\substack{l,m\in\{0,1,\ldots,B-1\} \\ l<m}} \prod_{k=0}^{K-1} \left|\Omega(\psi_B(n_k, l)) - \Omega(\psi_B(n_k, m))\right|^2 \right]. \quad (2.20)$$

To find γ^*_{min}, we only have to search over $(B-1)^K$ possibilities. In the case of the $B = 16$, $K = 5$ example, the size of the search space will be less than $8 \cdot 10^5$. Once

the $n_0^*, n_1^*, \ldots, n_{K-1}^*$ parameter values that maximize the minimum determinant have been calculated, the channel symbol indices can be determined as

$$i_l^k = \psi_B(n_k^*, l), \qquad k = 0, 1, \ldots, K-1, \qquad l = 0, 1, \ldots, B-1. \tag{2.21}$$

2.5 Code Design Examples

This section will provide some examples that demonstrate the code design method described in the previous sections. First, we will describe relationship between the repetition coded delay diversity scheme of [60] and our approach. The delay diversity scheme was chosen as a basis for comparison because, to our knowledge, this is the only existing procedure that can be used to construct ST trellis codes for any number of transmit antennas and any constellation. Then, we will give specific code design examples.

The delay diversity scheme is a special case of our design method. For $B = 2$ (e.g. the BPSK constellation) and $N = N_{min}$, the two methods are equivalent: the design rules for full spatial diversity uniquely determine the code. For $B > 2$, the delay diversity scheme corresponds to $n_k^* = 1$ ($k = 0, 1, \ldots, K-1$), which leads to $i_l^k = l$ ($k = 0, 1, \ldots, K-1$, $l = 0, 1, \ldots, B-1$). In [60], it was shown that, if Δ denotes the minimum Euclidean distance of the chosen constellation, then the minimum determinant of the resulting delay diversity ST code will be $\gamma_{min}^D = \Delta^{2K}$.

To characterize the theoretical performance improvement of our method over the delay diversity ST codes, we will use the *relative coding advantage*, β, defined as $\beta = \sqrt[K]{\gamma_{min}^*/\gamma_{min}^D}$. This quantity describes the normalized vertical shift between the two error performance curves as the SNR becomes large.

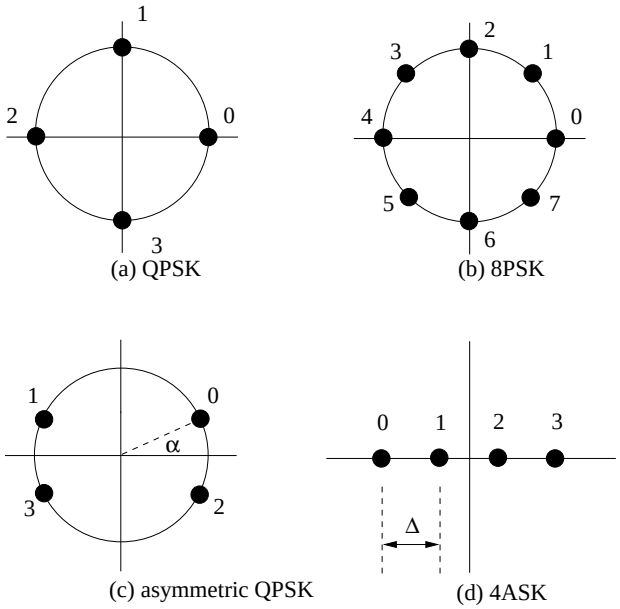

Figure 2.4: Example constellations

2.5.1 Code Design for B-ary PSK

In case of B-ary PSK modulation, the squared distance between constellation points l and m ($l, m \in \{0, 1, \ldots, B-1\}$) is given by

$$d^2(l,m) = |\Omega(l) - \Omega(m)|^2 = 4\sin^2\left(\frac{(l-m)\pi}{B}\right). \tag{2.22}$$

A pictorial representation of the QPSK and 8PSK constellations are shown in Figure 2.4 (a) and (b). Using (2.22) to express the minimum determinant of the code, the optimization problem becomes

$$\gamma^*_{min} = \max_{\substack{n_0,n_1,\ldots,n_{K-1} \\ \in\{1,2,\ldots,B-1\}}} \left[\min_{\substack{l,m\in\{0,1,\ldots,B-1\} \\ l<m}} 4^K \prod_{k=0}^{K-1} \sin^2\left(\frac{(\psi_B(n_k,l) - \psi_B(n_k,m))\pi}{B}\right)\right]. \tag{2.23}$$

If this optimization procedure is used for 3 antennas ($K = 3$) with QPSK ($B = 4$, $N = 16$), and Table 2.2 is used to generate the parametric permutation function values, then the result of the optimization will be $n_0^* = 1$, $n_1^* = 2$

and $n_2^* = 3$ with $\gamma_{min}^* = 16$. Note that this maximum is not unique: several other sets of $\{n_k^*\}$ values exist. This is not surprising because of the symmetry of the QPSK constellation and the commutativity of multiplication. The obtained permutations are: $(i_0^0, i_1^0, i_2^0, i_3^0)$=$(0, 1, 2, 3)$, $(i_0^1, i_1^1, i_2^1, i_3^1)$= $(0, 2, 3, 1)$, and $(i_0^2, i_1^2, i_2^2, i_3^2)$=$(0, 3, 1, 2)$. These permutations generate the ST code example depicted in Figure 2.1. The minimum determinant of the corresponding delay diversity ST code is $\gamma_{min}^D = \Delta^6 = 8$, resulting in a relative coding advantage of $\beta = 1.26$. For this particular case, the authors of [63] found a better code ($\gamma_{min} = 32$) through computer search. However, exhaustive search cannot be used to find good ST codes for a larger number of transmit antennas and larger constellation sizes because of its computational complexity.

For easy description, we need to find an efficient and concise representation for our ST codes. In [64], a generator matrix based approach was used. However, some of our codes belong to a more general class of ST codes because they cannot be described by generator matrices. To see this, consider the ST code example of Figure 2.1. The constellation point for antenna 2 is determined by the 2 most significant bits of the 4 bit state information. Let us denote these two bits by s_2 and s_3 ($s_2, s_3 \in \{0, 1\}$). In order to be able to describe this ST code by a generator matrix, the function $F(s_2, s_3)$, defined as

$$F(s_2, s_3) = (a_2 s_2 + a_3 s_3) \bmod 4,$$

should generate the indices 0,3,1,2 in this order for some a_2 and a_3 ($a_2, a_3 \in \{0, 1, 2, 3\}$). We trivially have $F(0, 0) = 0$. The $F(1, 0) = 3$ relation forces a_2 to be 3. Similarly, a_3 has to be 1, as a consequence of $F(0, 1) = 1$. Finally, the function will result in $F(1, 1) = 0$, which is not the desired value 2. Therefore, this ST code cannot be put in a generator matrix form. The possibly large number of encoder

states prevents us from using the trellis diagram, so the channel symbol index permutations will be used to describe the ST codes. Note that this representation is unique, and due to the regular structure of the proposed ST codes, it is easy to design simple encoders with $O(KB)$ hardware complexity.

Using the same procedure, ST codes for 3 transmit antennas ($K = 3$), 8PSK ($B = 8, N = 64$) and 16PSK ($B = 16, N = 256$) constellations have also been constructed. In the 8PSK case, the channel symbol index permutations

$$(i_0^0, i_1^0, i_2^0, i_3^0, i_4^0, i_5^0, i_6^0, i_7^0) = (0, 1, 2, 3, 4, 5, 6, 7)$$
$$(i_0^1, i_1^1, i_2^1, i_3^1, i_4^1, i_5^1, i_6^1, i_7^1) = (0, 1, 2, 3, 4, 5, 6, 7)$$
$$(i_0^2, i_1^2, i_2^2, i_3^2, i_4^2, i_5^2, i_6^2, i_7^2) = (0, 2, 4, 6, 3, 1, 7, 5)$$

are one of the possible sets of permutations that maximize the objective function, yielding the minimum determinant $\gamma_{min}^* = 0.6863$. The minimum determinant of the delay diversity scheme with the same design parameters is $\gamma_{min}^D = 0.2010$. These coding advantage values yield $\beta = 1.51$. For the 16PSK constellation, our design method resulted in the ST code given by the channel symbol indices

$$(i_0^0, i_1^0, i_2^0, i_3^0, i_4^0, i_5^0, i_6^0, i_7^0 i_8^0, i_9^0, i_{10}^0, i_{11}^0, i_{12}^0, i_{13}^0, i_{14}^0, i_{15}^0) =$$
$$= (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$$
$$(i_0^1, i_1^1, i_2^1, i_3^1, i_4^1, i_5^1, i_6^1, i_7^1 i_8^1, i_9^1, i_{10}^1, i_{11}^1, i_{12}^1, i_{13}^1, i_{14}^1, i_{15}^1) =$$
$$= (0, 2, 4, 6, 8, 10, 12, 14, 3, 1, 7, 5, 11, 9, 15, 13)$$
$$(i_0^2, i_1^2, i_2^2, i_3^2, i_4^2, i_5^2, i_6^2, i_7^2 i_8^2, i_9^2, i_{10}^2, i_{11}^2, i_{12}^2, i_{13}^2, i_{14}^2, i_{15}^2) =$$
$$= (0, 4, 8, 12, 3, 7, 11, 15, 6, 2, 14, 10, 5, 1, 13, 9).$$

The minimum determinant of this code is $\gamma_{min}^* = 0.110105$, while the delay diversity construction gives $\gamma_{min}^D = 0.003529$. The relative coding advantage is $\beta = 3.15$.

K	N	permutations	γ^*_{min}	γ^D_{min}	decoding depth
2	4	P_1, P_1	4	4	25
4	64	P_1, P_1, P_2, P_3	32	16	25
6	1024	$P_1, P_1, P_2, P_2, P_3, P_3$	256	64	25
8	2^{14}	$P_1, P_1, P_1, P_1,$ P_2, P_2, P_3, P_3	1024	256	35
10	2^{18}	$P_1, P_1, P_1, P_1, P_2,$ P_2, P_2, P_3, P_3, P_3	8192	1024	45

Table 2.3: ST codes for QPSK modulation

The reduced computational complexity of the proposed design procedure allowed us to construct ST codes for a large number of transmit antennas. Table 2.3 contains the brief description of the codes designed for QPSK constellation. The symbols P_1, P_2 and P_3 denote the permutations (0,1,2,3), (0,2,3,1) and (0,3,1,2), respectively. The permutations are assigned to transmit antennas from left to right. According to this notation, the 3 antenna ST code example of Figure 2.1 can be described as: P_1, P_2, P_3. The γ^*_{min} and γ^D_{min} values are also shown.

2.5.2 Code Design for Asymmetric QPSK

The next two examples employ constellations that are not used in current wireless communication systems, but they can illustrate the flexibility of the proposed design method. The first ST code was constructed for 4 transmit antennas ($K = 4$) and asymmetric QPSK modulation ($B = 4, N = 64$). The pictorial representation of the asymmetric QPSK constellation can be observed in Figure 2.4 (c). The parameter α, which is the angle between the signal points and the real axis, was set to $\pi/8$ (rad). The Euclidean distances between two arbitrary constellation points cannot be expressed in a closed form, but they can be easily calculated. The minimum distance of the constellation is $\Delta = 2\sin\alpha = 0.7654$. The optimization

procedure results in

$$(i_0^0, i_1^0, i_2^0, i_3^0) = (0, 1, 2, 3)$$

$$(i_0^1, i_1^1, i_2^1, i_3^1) = (0, 1, 2, 3)$$

$$(i_0^2, i_1^2, i_2^2, i_3^2) = (0, 2, 3, 1)$$

$$(i_0^3, i_1^3, i_2^3, i_3^3) = (0, 3, 1, 2)$$

permutations with $\gamma^*_{min} = 4.6863$. The minimum determinant of the delay diversity scheme is $\gamma^D_{min} = 0.1177$, so a relative coding advantage of $\beta = 2.51$ is achieved.

2.5.3 Code Design for 4ASK

We also designed a ST code for 4 transmit antennas ($K = 4$), and 4ASK constellation ($B = 4, N = 64$), shown in Figure 2.4 (d). The minimum distance of the normalized constellation was $\Delta = \sqrt{4/5}$. The squared distance between constellation points l and m ($l, m \in \{0, 1, 2, 3\}$) can be expressed as:

$$d^2(l, m) = |\Omega(l) - \Omega(m)|^2 = \Delta^2(l - m)^2.$$

The ST code design method found the same permutations as for the asymmetric QPSK case. The minimum determinant of the code is $\gamma^*_{min} = 1.6384$, and the delay diversity method yields $\gamma^D_{min} = 0.4096$. The resulting relative coding advantage is $\beta = 1.41$.

2.5.4 Discussion

The definition of the relative coding advantage allows us to predict the performance improvement before performing any simulation. If we compare the β values in the 3 antenna case for B-ary PSK, we can see that as the number of constellation

points (B) increases, the relative coding advantage also increases, and, therefore, more significant improvement is expected.

Based on the relative coding advantage values of the 4 antenna ST codes for QPSK, 4ASK and asymmetric QPSK modulations, improvement comparison can be made for a fixed constellation size ($B = 4$). The β values suggest that the proposed QPSK ST code will perform a little better than the delay diversity scheme, and the improvement will be more pronounced in the case of the 4ASK codes. Finally, the asymmetric QPSK ST code (whose actual performance depends on the value of α) seems to offer the largest improvement.

Due to the structure of the proposed ST codes, the minimum determinants are functions of product distances. The code design method tries to assign channel symbol indices to antennas at different states in such a way that the minimum value of the product distances is as large as possible. The minimum determinant of the delay diversity construction is only a function of the minimum distance of the constellation. Therefore, if the maximum distance of the chosen constellation is much larger than the minimum distance, our design method can exploit the additional degrees of freedom effectively, producing ST codes that perform much better than the delay diversity scheme. On the other hand, if the distances in the constellation have similar magnitudes, the proposed design method may not result in significant improvement.

2.6 Simulation Results

To illustrate the performance of the codes designed using the above described method, we show some simulation results. We compare our approach with the delay diversity scheme of [60], since, to our knowledge, this is the only method

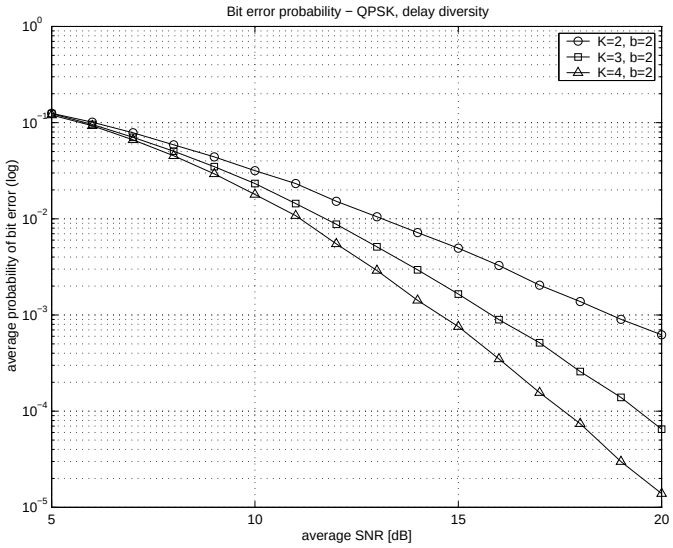

Figure 2.5: Delay diversity scheme with QPSK

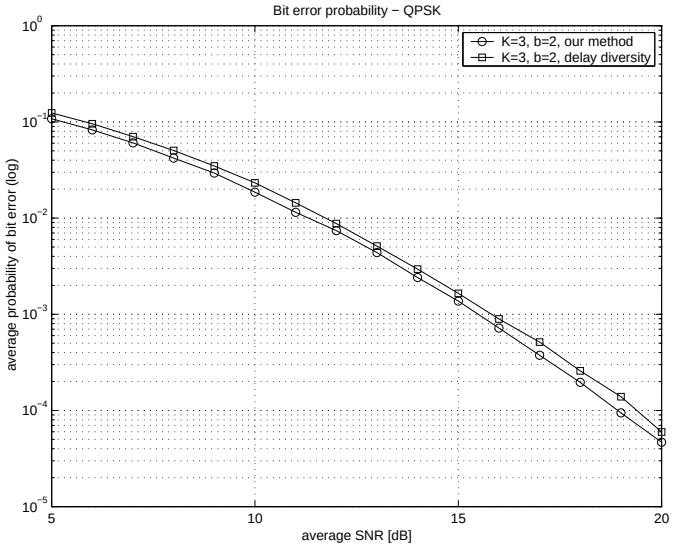

Figure 2.6: 3 transmit antennas with QPSK

that can be used to to construct ST codes for any number of transmit antennas and any memoryless constellation. The simulated communication system had one receive antenna. The source symbols were transmitted in frames of length 130, and the Viterbi algorithm (ML sequence detection) with decoding depth of 20 state transitions was used to decode the received signals. For each frame, the path gains between the transmit antennas and the receive antenna were modeled as independent, complex, zero mean, circularly symmetric Gaussian random variables with unit variance.

Since the frame error probability depends on the length of the frame and it does not seem very informative, we present probability of bit error curves as functions of the average signal to noise ratio (SNR) per source symbol at the receive antenna. In the sequel, the expression *coding gain* will refer to the difference (in dB) of transmit energies to achieve the same probability of bit error value. Both the coding advantage and the coding gain give information about the performance improvement, but the coding advantage is a theoretical quantity characterizing the *vertical* shift of the error performance curve, while the coding gain is experimental and it describes the *horizontal* shift.

The repetition coded delay diversity of [60] is a special case of our design rules. Figure 2.5 shows the performance of this scheme for different number of transmit antennas ($K = 2, 3, 4$ and $N = 4, 16, 64$, respectively) and QPSK modulation. We observe that the codes indeed provide different spatial diversity advantages since the steepness of the bit error rate curves is different. The rest of the figures compare the performance of the delay diversity construction and our approach using the example codes described in the previous section. Figure 2.6 depicts the results for 3 transmit antennas and QPSK modulation. The two probability of bit

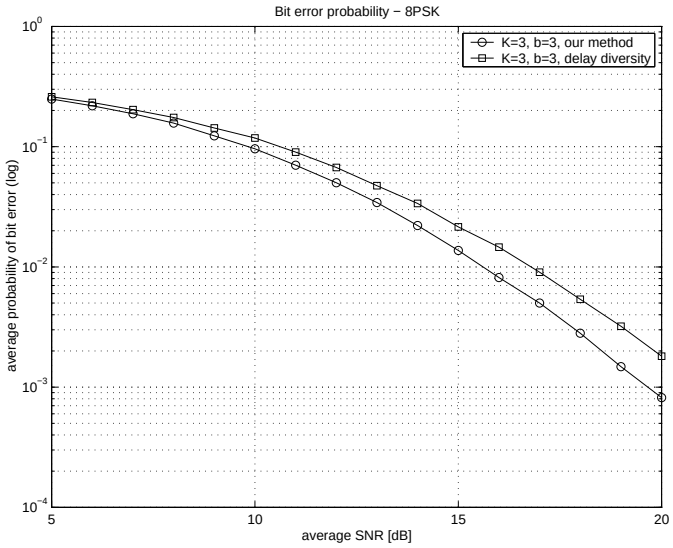

Figure 2.7: 3 transmit antennas with 8PSK

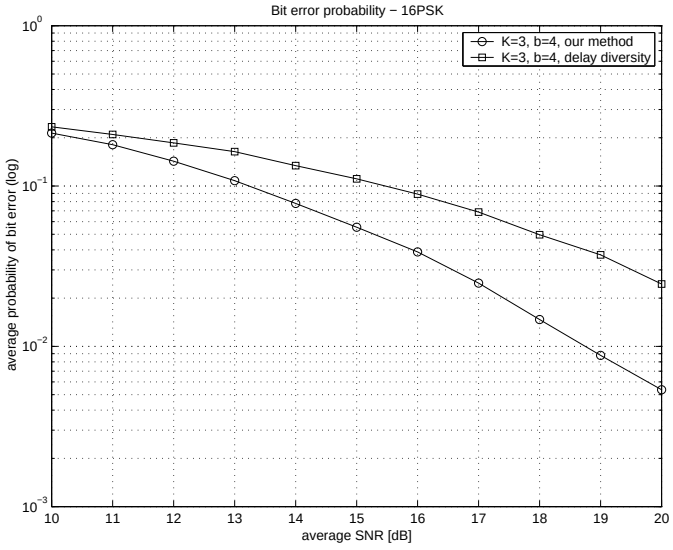

Figure 2.8: 3 transmit antennas with 16PSK

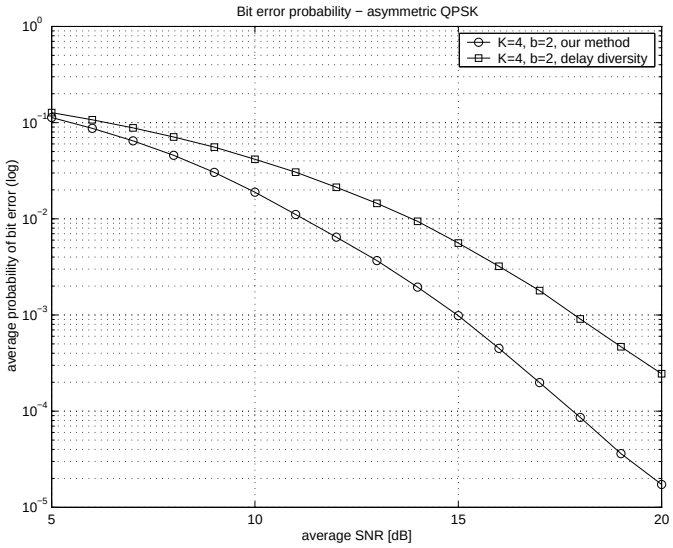

Figure 2.9: 4 transmit antennas with asymmetric QPSK

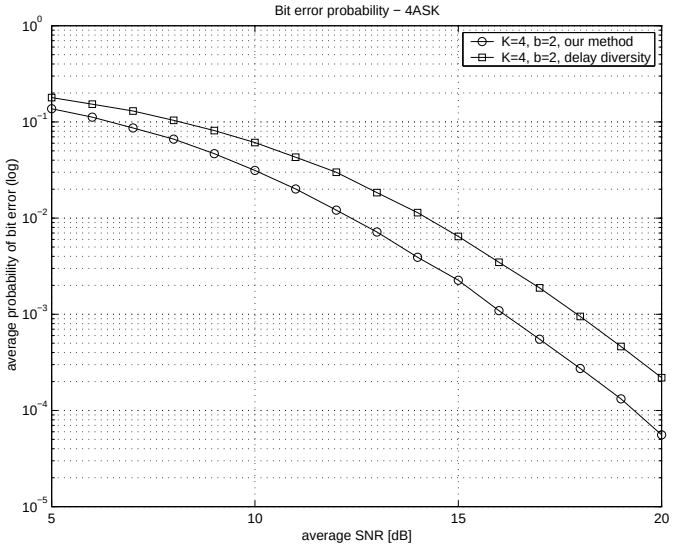

Figure 2.10: 4 transmit antennas with 4ASK

error curves are shifted versions of each other, as expected. Approximately 0.4-0.5 dB coding gain is observed over the delay diversity scheme.

Figure 2.7 shows the bit error rate curves of a 3 transmit antenna system with 8PSK modulation. At low SNR, the two error performance curves are close to each other, and they behave according to the expectations at medium and high SNRs. This phenomenon can be explained as follows. The definition of coding advantage [59] is based on an upper bound on the $Q(x)$ Gaussian tail probability function, and this bound is loose at low SNR. Moreover, the large number of transmission errors and the small minimum distance of the constellation may prevent the Viterbi algorithm with finite decoding depth from working properly at low SNR. The simulation shows that the performance improvement is more pronounced; at higher SNR, more than 1 dB coding gain can be achieved.

The performance of the ST code for 3 antennas and 16PSK constellation can be observed in Figure 2.8. Our ST code yields 2-2.5 dB coding gain compared to the delay diversity scheme. Figures 2.9 and 2.10 depict the bit error rate curves for the 4 antenna ST codes using asymmetric QPSK and 4ASK modulation, respectively. The first figure shows approximately 3 dB coding gain from medium SNR, and the second figure demonstrates 2 dB improvement over the delay diversity construction.

Since the number of states is exponential in the number of transmit antennas, it is not possible to decode the ST trellis codes designed for a large number of transmit antennas using ML sequence detection (Viterbi algorithm). Therefore, we chose a suboptimal tree decoding algorithm developed for convolutional coding and trellis coding: the M-algorithm [7]. This algorithm uses a tree structure to evaluate the metrics (in our case: the Euclidean distances) for the allowable channel symbol sequences. At each stage, it keeps at most M partial paths with the best metrics.

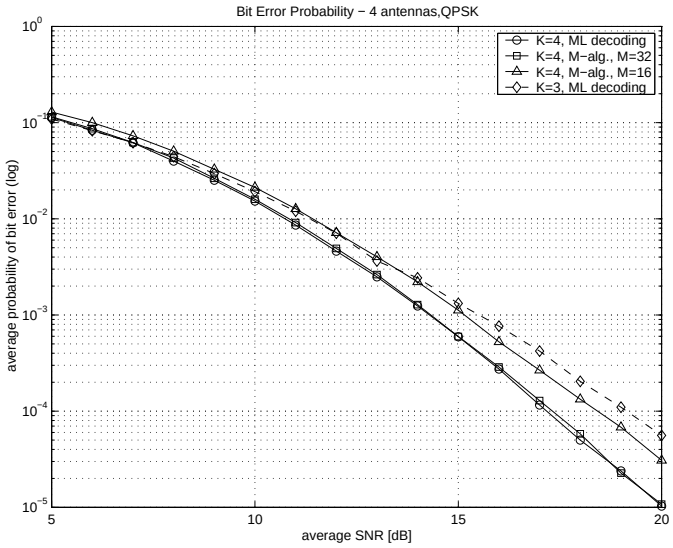

Figure 2.11: 4 antennas, QPSK, suboptimal decoding

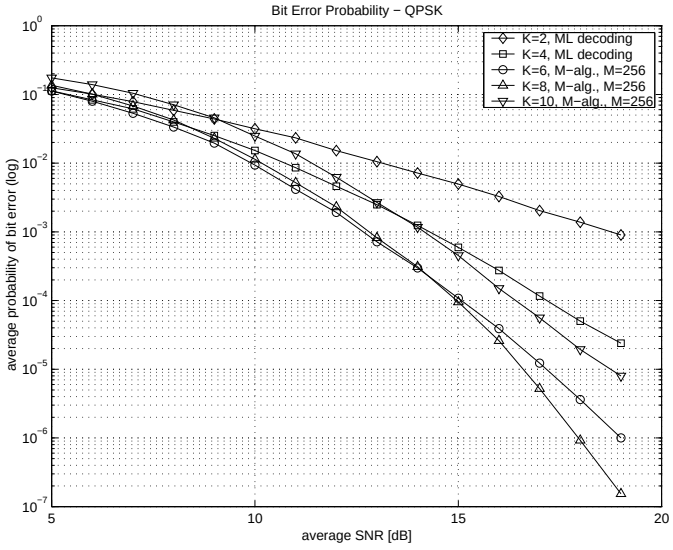

Figure 2.12: ST codes for QPSK modulation

Thus, the decoding complexity is $O(M)$, which is independent of the number of encoder states.

Figure 2.11 shows the performance of the 4 antenna, QPSK ST code for ML decoding and for suboptimal decoding. The bit error curve of the 3 antenna, QPSK ST code with ML decoding is also included for comparison. It can be observed that reducing the computational complexity (reducing the value of M) results in performance degradation.

Finally, the simulation results for the ST codes described in Table 2.3 are depicted in Figure 2.12. For 2 and 4 antennas, the ML decoding algorithm was used, and for the 6, 8 and 10 antenna cases, the ST codes were decoded by the suboptimal M-algorithm. The decoding complexity was kept approximately constant by setting $M = 256$. The used decoding depth values can be found in the last column of Table 2.3. The results show that as the N/M ratio increases, the performance loss increases.

2.7 Chapter Summary

Having observed the group/subgroup structure of the state transitions, we proposed systematic design rules for ST trellis codes that achieve full spatial diversity. For encoders having N_{min} states, we developed a code construction method that allows for ST code design for both diversity advantage and coding advantage. Due to the low complexity of the proposed design method, ST codes for a large number of antennas were also constructed. Based on the theoretical coding advantage values and the simulation results, we can make the following observations.

First, if the ratio of the maximum and the minimum distances of the chosen constellation is large, the additional optimization for coding advantage results in

codes that substantially outperform the codes that were designed only for diversity advantage.

Second, if the M-algorithm is used to decode the ST codes with constant decoding complexity in a quasi-static Rayleigh fading environment, increasing the number of transmit antennas will provide diminishing returns. The choice of the number of transmit antennas will depend on the actual allowed maximum computational complexity.

Chapter 3

Systematic Space-Time Trellis Code Design for Correlated MIMO Channels

3.1 Introduction

Most of the existing ST trellis code construction methods have assumed ideal channel models: either quasi-static fading or fast fading, without considering the effect of spatial or temporal correlation. For the quasi-static channel model, the authors of [59] proposed design rules for two transmit antennas to achieve the maximum achievable diversity advantage. Later works [60], [61], and Chaper 2 of this thesis described systematic code design methods for an arbitrary number of transmit antennas.

The first ST trellis code construction method for the fast fading channel model was described in [66]. ST codes for 2 transmit antennas and QPSK modulation were designed using the idea of signal set partitioning. In [67], the design of ST

codes for fast fading channels was also considered. The authors found ST codes for 2 transmit antennas and QPSK and 8PSK modulations through computer search.

For the quasi-static channel model, the authors of [68] investigated the achievable diversity order as a function of spatial correlation, taking into account some physical propagation parameters. The problem of code design for correlated fading channels was addressed in [70], and general performance criteria were derived for space-time-correlated Rayleigh fading channels. In [71], it was assumed that the channel stays constant for a number of channel symbol periods equal to the number of transmit antennas. The performance criteria were obtained for this channel model, and hand crafted trellis codes were proposed combining multiple trellis coded modulation with Alamouti's scheme [32].

In [72], characterizing the performance of space-time codes over space-time-correlated Rayleigh fading channels was also considered. The minimum diversity order achievable over all space-time correlation matrices of a given rank was defined as the measure of robustness. The relationship between the robustness (diversity) and the rank of the space-time correlation matrix was investigated, and upper and lower bounds on the achievable diversity were derived. However, exact results on the achievable diversity advantage and results on the coding advantage were not presented.

In this chapter, we consider the problem of ST trellis code design, taking into account both spatial and temporal channel correlation. As ad-hoc code design methods and computer search methods can only provide point-solutions in the design space, and the computational complexity of exhaustive search becomes prohibitive as the number of transmit antennas and the constellation size increase, we focus our attention on the development of *systematic* ST code design methods

that are flexible and scalable.

First, we derive the performance criteria for a channel model in which the channel changes from channel symbol period to channel symbol period in a correlated manner, assuming that the space-time correlation matrix is of full rank. We show that for this transmission scenario, the effect of the channel correlation and the ST code on the performance can be separated, resulting in channel-independent performance criteria. Our result implies that as long as the correlation matrix is of full rank, it does not matter what the correlation matrix actually is from the viewpoint of code design, and the ST code design problem for correlated channels can be reduced to the code design problem for fast fading channels. We characterize the performance of the ST codes by finding exact expressions for the achieved diversity advantage and coding advantage.

Then, we propose a new design criterion and analyze the properties of the ST trellis codes satisfying the proposed criterion. Finally, we develop a systematic code construction procedure that jointly considers diversity advantage and coding advantage for an arbitrary number of transmit antennas and any memoryless modulation. The proposed design method has extremely low complexity: we provide a closed form solution to the code design problem.

The rest of the chapter is organized in the following way. Section 3.2 introduces the mathematical model of the communication system and the notation. The performance criteria is derived in Section 3.3. Section 3.4 describes the proposed design criterion and its properties. The ST code design method is developed in Section 3.5. Section 3.6 contains the discussion, and Section 3.7 provides the simulation results. The summary is provided in the last section.

3.2 System Model and Notation

Consider a wireless communication system with K transmit and L receive antennas (the transmit antennas are indexed by $k,\ k \in \{0, 1, \ldots, K-1\}$, and the receive antennas are indexed by $l,\ l \in \{0, 1, \ldots, L-1\}$). The input bit stream is divided into b_s bit long blocks, forming B-ary ($B = 2^{b_s}$) source symbols. At discrete time t ($t = 0, 1, \ldots, T-1$), the ST encoder takes the current source symbol, b_t ($b_t \in \{0, 1, \ldots, B-1\}$), and outputs K B-ary channel symbol indices. We denote the channel symbol index for antenna k at time t by i_t^k. The channel symbol index vector takes the form: $\mathbf{i}_t = [\, i_t^0, i_t^1, \ldots, i_t^{K-1}\,]^T$. The channel symbol indices are mapped onto channel symbols (or constellation points) by the modulators and transmitted through the transmit antennas. $\Omega(i)$ will represent the constellation point corresponding to channel symbol index i (For example, in case of B-ary PSK, $\Omega(i) = \exp(j2\pi i/B)$, where $j = \sqrt{-1}$.). All the constellations are assumed to be normalized so that the average energy of the constellation is unity (if the channel symbols are equally likely). $c_t^k = \Omega(i_t^k)$ will denote the constellation point output by antenna k at time t. We will also use the channel symbol vector, defined as: $\mathbf{c}_t = [\, c_t^0, c_t^1, \ldots, c_t^{K-1}\,]^T$.

The transmission medium is assumed to be a flat (frequency non-selective), correlated Rayleigh fading channel. $\alpha_{k,l}(t)$ will represent the path gain between transmit antenna k and receive antenna l at time t. These path gains are modeled as complex, zero mean, Gaussian random variables with unit variance, and are assumed to be known by the receiver. Based on the above assumptions, after down-conversion, matched filtering and sampling, r_t^l, the received signal at receive

antenna l at discrete time t, can be expressed as

$$r_t^l = \sum_{k=0}^{K-1} \sqrt{\frac{\rho}{K}} \alpha_{k,l}(t) c_t^k + z_t^l. \tag{3.1}$$

The receiver noise, denoted by z_t^l, is taken from samples of independent, complex, zero mean, Gaussian random variables with unit variance. The average SNR per source symbol at receive antenna l will be defined as $SNR_l = \rho$.

Due to decoding errors, the receiver may decode a different sequence of channel symbols. The erroneously decoded channel symbol for transmit antenna k at time t will be denoted by $\hat{c}_t^k$, and the vector of decoded channel symbols at time t will be given by $\hat{\mathbf{c}}_t = [\, \hat{c}_t^0, \hat{c}_t^1, \ldots, \hat{c}_t^{K-1} \,]^T$.

In the sequel, the notation $\mathrm{diag}(a_1, a_2, \ldots, a_N)$ will be used to represent a diagonal matrix with scalar elements $a_1, a_2, \ldots, a_N$ along the main diagonal. The entries in the vectors, and the rows and columns of the matrices will be indexed from 0. All vectors are assumed to be column vectors, unless mentioned otherwise.

3.3 Performance Criteria

In this section, we derive the performance criteria for space-time correlated Rayleigh fading channels. The criteria are based on an upper bound on the pairwise error probability [70], derived for a general transmission scenario in which the received signal vector can be expressed as

$$\mathbf{r} = \sqrt{\frac{\rho}{K}} \mathbf{\Gamma} \alpha + \mathbf{z}. \tag{3.2}$$

In (3.2), $\mathbf{\Gamma}$ denotes the matrix of sent channel symbols, α stands for the complex, zero mean, Gaussian path gain vector with correlation matrix $\mathbf{R} = E(\alpha \alpha^{\mathcal{H}})$, and $\mathbf{z}$ denotes the receiver noise vector consisting of complex, zero mean, independent,

Gaussian random variables unit variance. It can be shown [70] that the probability that the maximum likelihood decoder erroneously decodes the channel symbol matrix $\hat{\mathbf{\Gamma}}$ if $\mathbf{\Gamma}$ was sent can be upper bounded as

$$P(\hat{\mathbf{\Gamma}}|\mathbf{\Gamma}) \leq \frac{\binom{2r-1}{r-1}\left(\frac{\rho}{K}\right)^{-r}}{\Pi_{i=1}^{r}\gamma_i}, \tag{3.3}$$

where r and γ_i's are the rank and the nonzero eigenvalues of the matrix $\mathbf{\Delta R \Delta}^{\mathcal{H}}$, respectively, and $\mathbf{\Delta}$ is the channel symbol difference matrix, defined as $\mathbf{\Delta} = \mathbf{\Gamma} - \hat{\mathbf{\Gamma}}$.

The performance criteria are obtained by evaluating (3.3) when the received signal is described by (3.1). Toward this end, we can define the matrix $\mathbf{\Gamma}^k = \text{diag}(c_0^k, c_1^k, \ldots, c_{T-1}^k)$, and the row vectors

$$\begin{aligned}
&\mathbf{r}^l = [r_0^l, r_1^l, \ldots, r_{T-1}^l], \\
&\alpha_{k,l} = [\alpha_{k,l}(0), \alpha_{k,l}(1), \ldots, \alpha_{k,l}(T-1)], \quad \text{and} \\
&\mathbf{z}^l = [z_0^l, z_1^l, \ldots, z_{T-1}^l].
\end{aligned}$$

Using these quantities, the LT by 1 received signal vector $\mathbf{r} = [\mathbf{r}^0, \mathbf{r}^1, \ldots, \mathbf{r}^{L-1}]^T$ is given by (3.2), with the LT by KLT channel symbol matrix

$$\mathbf{\Gamma} = \begin{bmatrix}
\mathbf{\Gamma}^0 & \mathbf{\Gamma}^1 & \ldots & \mathbf{\Gamma}^{K-1} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{\Gamma}^0 & \mathbf{\Gamma}^1 & \ldots & \mathbf{\Gamma}^{K-1} & \ldots & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\
 & & & & & & \ldots & & & & & & \\
\mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \ldots & \mathbf{\Gamma}^0 & \mathbf{\Gamma}^1 & \ldots & \mathbf{\Gamma}^{K-1}
\end{bmatrix}, \tag{3.4}$$

the LT by 1 noise vector $\mathbf{z} = [\mathbf{z}^0, \mathbf{z}^1, \ldots, \mathbf{z}^{L-1}]^T$, and the KLT by 1 path gain vector

$$\alpha = [\, \alpha_{0,0}, \alpha_{1,0}, \ldots, \alpha_{K-1,0}, \alpha_{0,1}, \alpha_{1,1}, \ldots, \alpha_{0,L-1}, \alpha_{1,L-1}, \ldots, \alpha_{K-1,L-1} \,]^T.$$

The correlation matrix $\mathbf{R}$ has KLT rows and KLT columns, and it is assumed to be of full rank (i.e. its eigenvalues are real and positive). Defining the matrix

$\hat{\mathbf{\Gamma}}^k = \text{diag}(\hat{c}_0^k, \hat{c}_1^k, \ldots, \hat{c}_{T-1}^k)$, the erroneously decoded channel symbol matrix, $\hat{\mathbf{\Gamma}}$, can be expressed similarly to (3.4), resulting in the LT by KLT channel symbol difference matrix $\mathbf{\Delta}$.

Assume that for τ time instants $t_0, t_1, \ldots, t_{\tau-1}$, the sent and the erroneously decoded channel symbol vectors are different, i.e. $\mathbf{c}_t - \hat{\mathbf{c}}_t \neq \mathbf{0}$ for $t \in \{t_0, t_1, \ldots, t_{\tau-1}\}$, and for the rest of the time instants, they are the same. Therefore, the sent and decoded channel symbol vectors corresponding to the times $t \notin \{t_0, t_1, \ldots, t_{\tau-1}\}$ will produce all zero rows and columns in the $\mathbf{\Delta}$ channel symbol difference matrix. These rows and columns can be eliminated from the analysis in the following way. For each $t \notin \{t_0, t_1, \ldots, t_{\tau-1}\}$, rows $t, t+T, t+2T, \ldots, t+(L-1)T$ and columns $t, t+T, t+2T, \ldots, t+(KL-1)T$ are removed from the matrix $\mathbf{\Delta}$, producing a new $L\tau$ by $KL\tau$ channel symbol difference matrix, $\mathbf{\Delta}'$. $\mathbf{\Delta}'$ has a structure similar to (3.4), but the matrices $\mathbf{\Gamma}^k$ are replaced with $\mathbf{\Delta}'^k = \text{diag}(c_{t_0}^k - \hat{c}_{t_0}^k, c_{t_1}^k - \hat{c}_{t_1}^k, \ldots, c_{t_{\tau-1}}^k - \hat{c}_{t_{\tau-1}}^k)$. Note that $\mathbf{\Delta}'$ has full row rank.

In addition, for each $t \notin \{t_0, t_1, \ldots, t_{\tau-1}\}$, rows and columns $t, t+T, t+2T, \ldots, t+(KL-1)T$ must also be removed from $\mathbf{R}$, resulting in the $KL\tau$ by $KL\tau$ matrix $\mathbf{R}'$. Since only all zero rows and columns have been deleted from $\mathbf{\Delta}$, the nonzero eigenvalues of $\mathbf{\Delta R \Delta}^{\mathcal{H}}$ and $\mathbf{\Delta}'\mathbf{R}'\mathbf{\Delta}'^{\mathcal{H}}$ are the same. It is shown in Appendix B that the relation

$$\prod_{i=1}^{r} \gamma_i = \det(\mathbf{\Delta}'\mathbf{R}'\mathbf{\Delta}'^{\mathcal{H}}) \geq \det(\mathbf{\Lambda}_{min}(L\tau)) \det(\mathbf{\Delta}'\mathbf{\Delta}'^{\mathcal{H}}) \tag{3.5}$$

holds, where $\mathbf{\Lambda}_{min}(L\tau)$ is a $L\tau$ by $L\tau$ diagonal matrix with the $L\tau$ smallest eigenvalues of $\mathbf{R}$ along the diagonal. Since $\mathbf{R}$ is positive definite, $\det(\mathbf{\Lambda}_{min}(L\tau))$ is strictly positive. Moreover, $\mathbf{\Delta}'$ has full row rank, so $\det(\mathbf{\Delta}'\mathbf{\Delta}'^{\mathcal{H}})$ is also strictly positive. Consequently, the matrices $\mathbf{\Delta R \Delta}^{\mathcal{H}}$ and $\mathbf{\Delta}'\mathbf{R}'\mathbf{\Delta}'^{\mathcal{H}}$ are both of rank $L\tau$.

Combining (3.3) with (3.5), and recognizing that

$$\det(\boldsymbol{\Delta}'\boldsymbol{\Delta}'^{\mathcal{H}}) = \prod_{i=0}^{\tau-1} ||\mathbf{c}_{t_i} - \hat{\mathbf{c}}_{t_i}||^{2L},$$

where $||\mathbf{x}|| = \sqrt{\mathbf{x}^{\mathcal{H}}\mathbf{x}}$, we arrive at the upper bound

$$P(\hat{\boldsymbol{\Gamma}}|\boldsymbol{\Gamma}) \leq \left(\frac{\rho}{K}\right)^{-L\tau} \binom{2L\tau - 1}{\tau - 1} \frac{1}{\det(\boldsymbol{\Lambda}_{min}(L\tau))} \prod_{i=0}^{\tau-1} ||\mathbf{c}_{t_i} - \hat{\mathbf{c}}_{t_i}||^{-2L}. \tag{3.6}$$

The performance criteria now can be formulated to minimize the maximum value of $P(\hat{\boldsymbol{\Gamma}}|\boldsymbol{\Gamma})$:

1. Design for diversity advantage (distance criterion): The minimum number of time instants when the correct and the decoded channel symbol vectors are different (the minimum value of τ) taken over all possible correct and erroneously decoded channel symbol vector sequences must be maximized.

2. Design for coding advantage (product criterion): The minimum of the norm products

$$\delta = \prod_{i=0}^{\tau-1} ||\mathbf{c}_{t_i} - \hat{\mathbf{c}}_{t_i}||^2$$

taken over all possible correct and erroneously decoded channel symbol vector sequences must be maximized.

Note that the these performance criteria are the same as the performance criteria proposed for fast (independently) fading channels [59]. This is not surprising since in case of independent fading, the matrices $\mathbf{R}$, $\mathbf{R}'$ and $\boldsymbol{\Lambda}_{min}(L\tau)$ become identity matrices, and (3.6) simplifies to a form similar to the upper bound derived in [59]. The relationship between the two upper bounds can be established by the inequality

$$\binom{2n-1}{n-1} < 4^n, \quad \text{for } n = 1, 2, 3, \ldots, \tag{3.7}$$

showing that (3.6) is tighter than the bound given in [59].

In the above derivation, the matrix $\mathbf{R}$ was assumed to have full rank. If the magnitudes of the correlation values $E\left[\alpha_{k_1,l_1}(t_1)\alpha^*_{k_2,l_2}(t_2)\right]$ diminish fast enough as $|k_1 - k_2|, |l_1 - l_2|$, and $|t_1 - t_2|$ increase, this assumption will be true. This corresponds to the condition that the magnitude of the correlation decays fast enough as the transmit and receive antenna separation and the time separation increase. If this condition holds, the space-time code design problem for correlated channels can be reduced to the code design problem for fast fading channels. Moreover, the correlation only causes coding advantage loss, and it is possible to achieve full diversity advantage. Here we define full diversity as the level of diversity achievable by a communication system having K transmit and L receive antennas operating in fast (independent) fading environment. However, if this condition is not satisfied, the correlation matrix may become rank deficient, causing loss of diversity advantage. In this case, the analysis can be performed by deleting more rows and columns from $\mathbf{R}$ and $\mathbf{\Delta}$. In this chapter, we assume that $\mathbf{R}$ is of full rank.

3.4 A Design Criterion for Trellis Codes

This section proposes a new design criterion that is based on the distance criterion described in the previous section. Since the design criterion is specific to trellis codes, it is necessary to extend the notation to explicitly show the dependence of the channel symbols on the state transitions.

The ST trellis encoder works as a finite state machine with N states: at discrete time t, it takes the current source symbol b_t, and governed by this input and the current state, S_t ($S_t \in \{0, 1, \ldots, N-1\}$), it moves to the next state, S_{t+1}. $i^k(S_t, b_t) = i^k_t$ will stand for the channel symbol index for transmit antenna k produced by the encoder during the state transition from S_t through the branch

corresponding to b_t. The channel symbol index vector $\mathbf{i}_t$ will become

$$\mathbf{i}(S_t, b_t) = [\, i^0(S_t, b_t), i^1(S_t, b_t), \ldots, i^{K-1}(S_t, b_t)\,]^T,$$

and the channel symbol vector $\mathbf{c}_t$ will be written as

$$\mathbf{c}(S_t, b_t) = [\, \Omega(i^0(S_t, b_t)), \Omega(i^1(S_t, b_t)), \ldots, \Omega(i^{K-1}(S_t, b_t))\,]^T.$$

Similarly, $\hat{\mathbf{c}}_t = \mathbf{c}(\hat{S}_t, \hat{b}_t)$ will be the erroneously decoded channel symbol vector at time t. This notation emphasizes that in general, the state sequences corresponding to the correct and the erroneously decoded paths (i.e. $\{S_t\}$ and $\{\hat{S}_t\}$) are different, and so are the encoded source symbol sequence $\{b_t\}$ and the decoded source symbol sequence $\{\hat{b}_t\}$.

The design criterion was developed for encoders having $N_{min} = B^{K-1}$ states, so from now on it will be assumed that the encoder has N_{min} states. The proposed design criterion (uniqueness criterion) is: *Every channel symbol index vector must be unique.* That is:

(a) The channel symbol index vectors assigned to different branches emanating from the same state must be different.

(b) Any channel symbol index vector assigned to any state must be different from any channel symbol index vector assigned to any other state.

Each channel symbol index vector contains K B-ary symbols, so there are B^K different channel symbol index vectors. Since there are B branches emanating from each state and the encoder is assumed to have N_{min} states, we need exactly B^K different channel symbol index vectors. Therefore, it is possible to assign channel symbol index vectors to state transitions according to the uniqueness criterion.

The following two subsections will analyze the properties of the ST codes that satisfy the proposed criterion.

3.4.1 Diversity Advantage

Assume that the first decoding error occurs at $t = t_0$, so the correct and decoded paths diverge at this point (i.e. $S_{t_0} = \hat{S}_{t_0}$ and $b_{t_0} \neq \hat{b}_{t_0}$). The two paths are assumed to merge later at time $t = t_0 + \tau$, resulting in a τ-long error event. As a consequence of criterion (a), the channel symbol vectors corresponding to the two paths diverging at S_{t_0} will be different. Moreover, as a result of criterion (b), the channel symbol vectors corresponding to the correct and the decoded paths going through different states will be different. Therefore, for $t = t_0, t_0+1, \ldots, t_0+\tau-1$, we have $\mathbf{c}(S_t, b_t) - \mathbf{c}(\hat{S}_t, \hat{b}_t) \neq \mathbf{0}$. From the performance criteria described in Section 3, one can conclude that any τ-long error event will achieve a diversity advantage of τL. In case of multiple error events, the total diversity advantage will be the sum of the diversity advantages of the individual error events. Consequently, the minimum diversity advantage of the ST code is determined by the shortest error event. It was established in Chapter 2 that for ST encoders having N_{min} states, the shortest error event is K state transitions long, so the ST code is guaranteed to achieve a diversity advantage of KL.

3.4.2 Error Event Probability

In this section, we will derive an upper bound on the τ-long error event probability, i.e. the probability that at a given discrete time instant, a decoding error occurs, resulting in an error event that is τ state transitions long.

Using the union bound, the τ-long error event probability, which is the probability of the union of the τ-long error events, can be upper bounded as

$$P_\tau \leq \sum_{\mathbf{\Gamma}} \sum_{\hat{\mathbf{\Gamma}}_\tau} P(\hat{\mathbf{\Gamma}}_\tau | \mathbf{\Gamma}) P(\mathbf{\Gamma}). \tag{3.8}$$

In (3.8), the first summation is over all possible sent channel symbol matrices $\mathbf{\Gamma}$, and the second summation is over all possible erroneously decoded channel symbol matrices $\hat{\mathbf{\Gamma}}_\tau$ corresponding to τ-long error events for a particular $\mathbf{\Gamma}$. Assuming that the decoding error occurs at time t_0, for $t = t_0, t_0+1, \ldots, t_0+\tau-1$, at least one coordinate of the channel symbol vectors $\mathbf{c}(S_t, b_t)$ and $\mathbf{c}(\hat{S}_t, \hat{b}_t)$ will be different. Therefore, if we denote the minimum distance of the chosen constellation by d, the squared norms of their differences can be lower bounded as

$$||\mathbf{c}(S_t, b_t) - \mathbf{c}(\hat{S}_t, \hat{b}_t)||^2 \geq d^2 \quad \text{for } t = t_0, t_0+1, \ldots, t_0+\tau-1. \tag{3.9}$$

Combining (3.6) with (3.7) and (3.9), we obtain an upper bound on the pairwise error probability:

$$P(\hat{\mathbf{\Gamma}}_\tau|\mathbf{\Gamma}) < \frac{1}{\det(\mathbf{\Lambda}_{min}(L\tau))} \left(\frac{\rho}{4K}\right)^{-L\tau} d^{-2L\tau}. \tag{3.10}$$

Defining γ_{min}, the minimum determinant taken over all possible τ values, as

$$\gamma_{min} = \min_{1\leq\tau\leq T-1} \det(\Lambda_{min}(L\tau)), \tag{3.11}$$

the pairwise error probability can be further upper bounded:

$$P(\hat{\mathbf{\Gamma}}_\tau|\mathbf{\Gamma}) < \frac{1}{\gamma_{min}} \left(\frac{\rho}{4K}\right)^{-L\tau} d^{-2L\tau}. \tag{3.12}$$

Since the upper bound (3.12) does not depend on $\hat{\mathbf{\Gamma}}_\tau$, (3.8) can be rewritten as:

$$P_\tau < \frac{1}{\gamma_{min}} \sum_{\mathbf{\Gamma}} N(\hat{\mathbf{\Gamma}}_\tau) \left(\frac{\rho d^2}{4K}\right)^{-L\tau} P(\mathbf{\Gamma}), \tag{3.13}$$

where $N(\hat{\mathbf{\Gamma}}_\tau)$ is the number of τ-long error paths for a given correct path. $N(\hat{\mathbf{\Gamma}}_\tau)$ can be strictly upper bounded by B^τ, so the upper bound simplifies to:

$$P_\tau < \frac{1}{\gamma_{min}} \sum_{\mathbf{\Gamma}} B^\tau \left(\frac{\rho d^2}{4K}\right)^{-L\tau} P(\mathbf{\Gamma}). \tag{3.14}$$

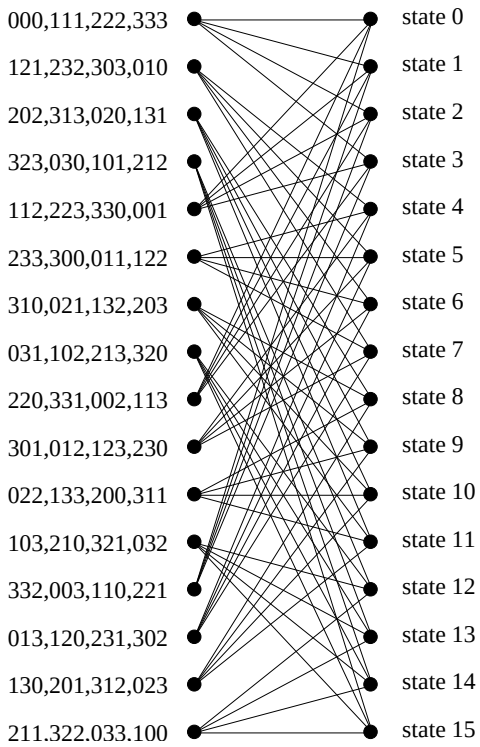

Figure 3.1: Example ST code for 3 antennas, 4-ary modulation

Finally, recognizing that $P(\mathbf{\Gamma})$ is the only term that depends on $\mathbf{\Gamma}$, and that the probability mass function of $\mathbf{\Gamma}$ sums to unity, the final expression for the upper bound on the node error probability becomes:

$$P_\tau < \frac{1}{\gamma_{min}} \left(\frac{\rho d^2}{4K \sqrt[L]{B}} \right)^{-L\tau}. \tag{3.15}$$

From (3.15), one can see that the probability that a τ-long error event occurs decreases at least exponentially with τ. Consequently, at high SNR, the shortest error events will dominate; the longer error paths will have negligible contribution to the error event probability. (Note that we assumed that the trellis is finite, so the range of the τ values is constrained by the length of the trellis, the time when the decoding error occurs and the length of the shortest error event.)

3.5 The Design Method

In the ST encoder, B channel symbol index vectors are assigned to each state, according to the branches emanating from that state. The current source symbol selects one of them, and the kth ($k = 0, 1, \ldots, K-1$) coordinate of the chosen vector determines the constellation point for antenna k. Figure 3.1 depicts an example ST code for 3 antennas and any 4-ary constellation ($K = 3$, $B = 4$, $N = 16$). In this case, if the current state is state 2 and the value of the current source symbol is 3, the ST encoder selects the 3rd channel symbol index vector, $[1, 3, 1]^T$, and moves to state 11. The 0th, lst and 2nd antennas will transmit the channel symbols corresponding to the indices 1, 3 and 1, respectively.

This section addresses the problem of assigning channel symbol index vectors to state transitions. The most important objective is to maximize the diversity advantage, so the ST codes are required to satisfy the uniqueness criterion. The remaining freedom can be used to increase the value of the minimum norm product (δ). Since the available B^K channel symbol index vectors can be arranged in $(B^K)!$ different ways, the computational complexity of exhaustive search becomes prohibitive, as the number of transmit antennas and the constellation size increase. Therefore, we propose an approach that does not guarantee optimality, but is simple and flexible. The basic idea behind the method is that it attempts to maximize the minimum norm product corresponding to the shortest error events by maximizing the number of nonzero coordinates in the channel symbol difference vectors, $\mathbf{c}(S_t, b_t) - \mathbf{c}(\hat{S}_t, \hat{b}_t)$. Since the shortest error events achieve the minimum diversity advantage, it is a reasonable objective to maximize the norm product of these error events. Moreover, it was shown in Section 3.4.2 that at higher SNR, the shortest error events will dominate the performance of the ST codes.

Since the ST encoder has B^{K-1} states, any state S ($S \in \{0, 1, \ldots, B^{K-1} - 1\}$) can be uniquely represented as a $K-1$ digit B-ary number with digits $l_1, l_2, \ldots, l_{K-1}$ ($l_k \in \{0, 1, \ldots, B-1\}$):

$$S = B^{K-2} l_{K-1} + B^{K-3} l_{K-2} + \ldots + B l_2 + l_1. \tag{3.16}$$

The proposed design rules are:

1. The 0th coordinate of the channel symbol index vector (the channel symbol index for the 0th transmit antenna) corresponding to input b ($b \in \{0, 1, \ldots, B-1\}$) at state S is determined as:

$$i^0(S, b) = (b + l_{K-1} + l_{K-2} + \ldots + l_1) \bmod B. \tag{3.17}$$

2. The rest of the coordinates (the channel symbol indices for the rest of the transmit antennas) are calculated as:

$$i^k(S, b) = (i^0(S, b) + l_k) \bmod B \quad \text{for } k = 1, 2, \ldots, K-1. \tag{3.18}$$

By making use of the identity

$$\left(\sum_{i=1}^{n} \alpha_i \bmod \beta \right) \bmod \beta = \left(\sum_{i=1}^{n} \alpha_i \right) \bmod \beta, \tag{3.19}$$

the second design rule can be put in the alternative form:

$$i^k(S, b) = (b + l_{K-1} + l_{K-2} + \ldots + l_1 + l_k) \bmod B. \tag{3.20}$$

Theorem 3.1: The ST codes produced by the above described construction method satisfy the uniqueness criterion for any B-ary constellation.

Proof: The proof is given in two steps.

(a) From (3.17), it can be seen that for a given state S (given $l_1, l_2, \ldots, l_{K-1}$ values) and different b input values, the value of $i^0(S, b)$ will be different. Therefore,

the channel symbol index vectors assigned to the same state are different in at least one (the 0th) coordinate.

(b) Let us pick two arbitrary states S_1 and S_2 ($S_1 \neq S_2$), and one channel symbol index vector assigned to a branch emanating from each state. If the 0th coordinates of the index vectors are different, then the statement is proven. If they are the same, then we have $i^0(S_1, b_1) = i^0(S_2, b_2)$ for some $b_1, b_2 \in \{0, 1, \ldots, B-1\}$. The states S_1 and S_2 can be uniquely expressed as

$$S_1 = B^{K-2} l_{K-1} + B^{K-3} l_{K-2} + \ldots + B l_2 + l_1$$
$$S_2 = B^{K-2} m_{K-1} + B^{K-3} m_{K-2} + \ldots + B m_2 + m_1,$$

with $l_k, m_k \in \{0, 1, \ldots, B-1\}$. $S_1 \neq S_2$, so $l_k \neq m_k$ must hold for at least one k value. Since the kth ($k = 1, 2, \ldots, K-1$) coordinates of the channel symbol index vectors are determined according to (3.18), we will have $i^k(S_1, b_1) \neq i^k(S_2, b_2)$ for at least one k value. Therefore, the channel symbol index vectors assigned to different states are different in at least one coordinate. □

As an example, consider the ST code shown in Figure 3.1. Since $B = 4$, state $S = 6$ can be represented as $S = Bl_2 + l_1$ with $l_2 = 1$ and $l_1 = 2$. The channel symbol indices corresponding to the 2nd branch ($b = 2$) emanating from state 6 are determined as:

$$i^0(S, b) = (b + l_2 + l_1) \bmod B = 1,$$
$$i^1(S, b) = (i^0(S, b) + l_1) \bmod B = 3,$$
$$i^2(S, b) = (i^0(S, b) + l_2) \bmod B = 2.$$

Therefore, the channel symbol index vector assigned to this state transition will be $[1, 3, 2]^T$.

The following theorem describes the number and the location of the nonzero coordinates in the channel symbol difference vectors corresponding to the shortest error events in case of the proposed design rules. According to the theorem, for each correct path, there are $B-2$ K-long error paths that produce as many nonzero coordinates as possible, and there is one K-long error path that produces one zero coordinate in $K-1$ channel symbol difference vectors.

Theorem 3.2: Assume that the correct and the erroneously decoded paths diverge at S_0 and merge at S_K (i.e. $S_0 = \hat{S}_0$, $b_0 \neq \hat{b}_0$, and $S_K = \hat{S}_K$). Then, for any $B \geq 2$, the channel symbol difference vector $\mathbf{c}(S_0, b_0) - \mathbf{c}(\hat{S}_0, \hat{b}_0)$ contains only nonzero entries. If $\hat{b}_0 = (b_0 + B/2) \bmod B$, then for $t = 1, 2, \ldots, K-1$, the tth coordinate of the vectors $\mathbf{c}(S_t, b_t) - \mathbf{c}(\hat{S}_t, \hat{b}_t)$ is zero, and the rest of the coordinates are nonzero. For each correct path, there is exactly one such K-long error path. The remaining $B-2$ K-long error paths will produce channel symbol difference vectors having only nonzero coordinates for $t = 1, 2, \ldots, K-1$.

Proof: First, consider the state transition $S_0 \rightarrow S_1$. In the proof of *Theorem 1*, it has been shown that the 0th coordinates of the channel symbol index vectors assigned to the same state are different, so we have $i^0(S_0, b_0) \neq i^0(\hat{S}_0, \hat{b}_0)$. Since $S_0 = \hat{S}_0$ and $b_0 \neq \hat{b}_0$, design rule (3.20) yields the relationship $i^k(S_0, b_0) \neq i^k(\hat{S}_0, \hat{b}_0)$ for $k = 1, 2, \ldots, K-1$. Therefore, the vector $\mathbf{c}(S_0, b_0) - \mathbf{c}(\hat{S}_0, \hat{b}_0)$ contains only nonzero entries.

The next step is to analyze the rest of the state transitions $S_t \rightarrow S_{t+1}$ for $t = 1, 2, \ldots, K-1$. In Chapter 2, a closed form expression for S_K was derived as a function of the starting state, S_0, and the input symbol sequence $b_1, b_2, \ldots, b_{K-1}$. If $N = N_{min}$, from (2.6) we have

$$S_K = B^{K-2}b_1 + B^{K-3}b_2 + \ldots + Bb_{K-2} + b_{K-1} = \sum_{m=1}^{K-1} B^{K-1-m}b_m. \qquad (3.21)$$

Since the error path is K state transitions long, we have $S_K = \hat{S}_K$. Equation (3.21) is a unique representation for S_K, so the relation $b_t = \hat{b}_t$ must hold for $t = 1, 2, \ldots, K-1$. Moreover, since the source symbol b_0 can take on B different values, there are exactly B paths that start at S_0 and end at S_K. This means that there are exactly $B-1$ K-long error paths for a given correct path. The state transition sequence for $t = 1, 2, \ldots, K-1$ was also described in Chapter 2 (Equation (2.5)) as a function of the starting state, S_0, and the input symbol sequence $b_0, b_1, \ldots, b_{t-1}$:

$$S_t = B^t \left(S_0 \bmod (B^{K-t-1}) \right) + \sum_{m=0}^{t-1} B^{t-1-m} b_m. \tag{3.22}$$

Comparing (3.22) and (3.16), we can express the state transitions corresponding to the correct and erroneously decoded paths:

$$S_t = B^{K-2} l_{K-1} + B^{K-3} l_{K-2} + \ldots + B^t l_{t+1} + B^{t-1} l_t + B^{t-2} l_{t-1} + \ldots + l_1$$

$$\hat{S}_t = B^{K-2} l_{K-1} + B^{K-3} l_{K-2} + \ldots + B^t l_{t+1} + B^{t-1} \hat{l}_t + B^{t-2} l_{t-1} + \ldots + l_1$$

for some $l_{K-1}, l_{K-2}, \ldots, l_{t+1}$ values, with $l_t = b_0$, $\hat{l}_t = \hat{b}_0$ and $l_k = b_{t-k} = \hat{b}_{t-k}$ for $k = 1, 2, \ldots, K-1$. In the above two expressions, only the terms l_t and $\hat{l}_t$ are different, so the design rule (3.17) ensures that $i^0(S_t, b_t) \neq i^0(\hat{S}_t, \hat{b}_t)$ for $t = 1, 2, \ldots, K-1$. Thus, the 0th coordinates of the vectors $\mathbf{c}(S_t, b_t) - \mathbf{c}(\hat{S}_t, \hat{b}_t)$ are nonzero for $t = 1, 2, \ldots, K-1$.

Using the notation $a_t = l_{K-1} + l_{K-2} + \ldots + l_{t+l} + l_{t-1} + \ldots + l_1$, the kth ($k = 1, 2, \ldots, K-1$) coordinates of the channel symbol index vectors corresponding to the correct and the erroneously decoded paths at time t ($t = 1, 2, \ldots, K-1$) will be

$$\begin{aligned} i^k(S_t, b_t) &= (b_t + a_t + l_t + l_k) \bmod B, \\ i^k(\hat{S}_t, \hat{b}_t) &= (\hat{b}_t + a_t + \hat{l}_t + l_k) \bmod B, \end{aligned} \tag{3.23}$$

as a consequence of design rule (3.20). If $k \neq t$, the only different terms in the summations of (3.23) are l_t and $\hat{l}_t$, so $i^k(S_t, b_t)) \neq i^k(\hat{S}_t, \hat{b}_t)$. If $k = t$, using that $l_t = b_0$ and $\hat{l}_t = \hat{b}_0$, the channel symbol indices in (3.23) can be rewritten as

$$\begin{aligned} i^k(S_t, b_t) &= (b_t + a_t + 2b_0) \bmod B, \\ i^k(\hat{S}_t, \hat{b}_t) &= (\hat{b}_t + a_t + 2\hat{b}_0) \bmod B. \end{aligned} \tag{3.24}$$

Since $b_t = \hat{b}_t$ for $t = 1, 2, \ldots, K-1$, $i^k(S_t, b_t)$ will be equal to $i^k(\hat{S}_t, \hat{b}_t)$ if and only if $|\hat{b}_0 - b_0|$ is a multiple of $B/2$. (Note that B is a power of 2, so $B/2$ is always integer and a power of 2.) Since b_0 and $\hat{b}_0$ take on values from the set $\{0, 1, \ldots, B-1\}$, this can only happen if $\hat{b}_0 = (b_0 + B/2) \bmod B$. For each correct path (each b_0), there is only one K state transitions long erroneous path ($\hat{b}_0$) with this property. In this case, for $t = 1, 2, \ldots, K-1$, the tth coordinate of the vectors $\mathbf{c}(S_t, b_t) - \mathbf{c}(\hat{S}_t, \hat{b}_t)$ will be zero, and the rest of the coordinates will be nonzero. For the remaining $B - 2$ K-long error paths, all the coordinates will be nonzero. □

The next theorem characterizes the performance of the 2 antenna PSK ST trellis codes designed by our code construction procedure in the case of the quasi-static fading channel model. It states that these codes also achieve full spatial diversity in quasi-static fading environment.

Theorem 3.3: The ST trellis codes designed by the proposed method for 2 transmit antennas and B-ary PSK modulation satisfy the rank criterion [59] derived for quasi-static fading channels.

Proof: Assume that the correct and the decoded paths diverge at state S_0 (i.e. $S_0 = \hat{S}_0$ and $b_0 \neq \hat{b}_0$). Since the shortest error event is K state transitions long, the 2 paths can only merge at state S_K or later. Assuming that the 2 paths merge at state S_{T+K} for some $T \geq 0$ and considering the $K = 2$ case, we have

$S_{T+2} = \hat{S}_{T+2}$. The basic idea behind the proof is to show that the channel symbol difference vectors corresponding to $t = 0$ (when the 2 paths diverge) and $t = T+2$ (when the 2 paths merge) form a linearly independent set. We will do this by proving that the matrix $\mathbf{D_2}$, defined as

$$\mathbf{D_2} = \begin{bmatrix} d_0^0 & d_{T+2}^0 \\ d_0^1 & d_{T+2}^1 \end{bmatrix}, \tag{3.25}$$

where $d_t^k = \Omega(i^k(S_t, b_t)) - \Omega(i^k(\hat{S}_t, \hat{b}_t))$, is of full rank for all possible $T+2$ long correct and erroneously decoded paths.

The 0th column of $\mathbf{D_2}$ can be determined easily, recognizing that the 2 antenna ST encoder has B states and expressing the first erroneously decoded source symbol, $\hat{b}_0$, as $\hat{b}_0 = (b_0 + \Delta_b) \bmod B$ for some $\Delta_b \in \{1, 2, \ldots, B-1\}$. Using design rules (3.17), (3.20) and the identity (3.19), and assuming B-ary PSK modulation, we obtain

$$\begin{aligned} d_0^0 &= e^{j\frac{2\pi}{B}(S_0+b_0)}\left(1 - e^{j\frac{2\pi}{B}\Delta_b}\right) \\ d_0^1 &= e^{j\frac{2\pi}{B}(2S_0+b_0)}\left(1 - e^{j\frac{2\pi}{B}\Delta_b}\right). \end{aligned} \tag{3.26}$$

Note that the modulo operation can be omitted because of the periodicity of the complex exponentiation.

The first column of $\mathbf{D_2}$ is determined as follows. It is shown in Appendix C that for encoders having N_{min}, states, the state transition $S_{T+K}, T \geq 0$, can be expressed as

$$S_{T+K} = \sum_{m=T+1}^{T+K-1} B^{T+K-1-m} b_m. \tag{3.27}$$

For $K = 2$, (3.27) simplifies to $S_{T+2} = b_{T+1}$. Since the correct and the erroneous paths merge at S_{T+2}, we have $S_{T+2} = b_{T+1} = \hat{S}_{T+2} = \hat{b}_{T+1}$. Moreover, the 2 paths do not merge before time $T+2$, so state $\hat{S}_{T+1}$ can be written as $\hat{S}_{T+1} =$

number of TX antennas	2	3	4
BPSK	16	64	256
QPSK	4	8	16
8PSK	0.3431	0.0345	0.0035
16PSK	0.0232	$2.4 \cdot 10^{-5}$	-
4ASK	0.6400	0.5120	0.4096

Table 3.1: Minimum determinant values

$(S_{T+1} + \Delta_S)$ mod B for some $\Delta_S \in \{1, 2, \ldots, B-1\}$. As a result, applying design rules (3.17), (3.20) and the identity (3.19), and assuming B-ary PSK modulation, we arrive at

$$\begin{aligned} d^0_{T+2} &= e^{j\frac{2\pi}{B}(S_{T+1}+b_{T+1})}\left(1 - e^{j\frac{2\pi}{B}\Delta_S}\right) \\ d^1_{T+2} &= e^{j\frac{2\pi}{B}(2S_{T+1}+b_{T+1})}\left(1 - e^{j\frac{2\pi}{B}2\Delta_S}\right). \end{aligned} \tag{3.28}$$

The matrix $\mathbf{D_2}$ is rank deficient if and only if its determinant is zero, i.e.

$$d^0_0 d^1_{T+2} = d^1_0 d^0_{T+2}. \tag{3.29}$$

Substituting (3.26) and (3.28) into (3.29) and simplifying the expression, the zero determinant condition becomes

$$e^{j\frac{2\pi}{B}S_{T+1}}\left(1 - e^{j\frac{2\pi}{B}2\Delta_S}\right) = e^{j\frac{2\pi}{B}S_0}\left(1 - e^{j\frac{2\pi}{B}\Delta_S}\right). \tag{3.30}$$

After taking squared magnitudes of both sides of (3.30) and simplification, we obtain

$$\cos\left(\frac{2\pi}{B}2\Delta_S\right) = \cos\left(\frac{2\pi}{B}\Delta_S\right). \tag{3.31}$$

Since B is a power of 2 and Δ_S takes on values from the set $\{1, 2, \ldots, B-1\}$, the two sides of (3.31) cannot be equal. Therefore, the determinant of $\mathbf{D_2}$ cannot be zero, so the ST code satisfies the rank criterion. □

We have not been able to prove this property for an arbitrary number of transmit antennas and any memoryless modulation, but we have verified it using computer simulations for some ST codes constructed by our method. Table 3.1 shows the minimum determinant values of the ST codes designed for 2, 3 and 4 transmit antennas, and BPSK, QPSK, 8PSK, 16PSK and 4ASK constellations. Since the values of the minimum determinants decay fast with the constellation size, the ST codes designed by previously existing methods [59], [60], [61], [63], and the method described in Chapter 2 may outperform the proposed ST codes in quasi-static fading environment. However, Theorem 3.3 and Table 3.1 suggest that if a temporally evolving channel becomes constant for a short time period (for example a vehicle stops at a red light), the proposed ST codes are still able to deliver the available (in this case only spatial) diversity.

3.6 Discussion

In order to achieve a diversity advantage of KL, there must be at least K time instants when the sent and the decoded channel symbol vectors are different. Thus, the shortest error event must be at least K state transitions long. This means that for a B-ary modulation (B branches emanating from each state), the encoder must have at least $N_{min} = B^{K-1}$ states. Consequently, our ST codes achieve the desired diversity level with the minimum possible trellis complexity.

Since the diversity does not depend on the dimensionality of the channel symbol vectors, it is possible to design ST codes that can achieve a diversity advantage of KL with encoders having less than K transmit antennas. However, our design criterion provides extra performance gain in addition to the achieved minimum diversity level. The uniqueness criterion guarantees that the probability of the

error events decreases exponentially with their lengths. For a ST code that does not satisfy this criterion, it is possible that the probability of a very long error event (many bit errors) and a very short error event (a few bit errors) are in the same order of magnitude, causing serious performance loss. On the other hand, the uniqueness criterion ensures that the probability of the long error events decays much faster than the probability of the short error events, as the SNR increases. From Section 4, it can be seen that for encoders having N_{min} states, in order to satisfy the uniqueness criterion, the channel symbol vectors must have at least K coordinates, so the encoder must have at least K transmit antennas.

3.7 Simulation Results

To illustrate the performance of the codes obtained by the proposed method, we present some simulation results. The source symbols were transmitted in frames of length 130, and the Viterbi algorithm with decoding depth of 20 state transitions was used to decode the received signals. For the fast fading channel model, the path gains between the transmit and the receive antennas were independent, complex, zero mean, Gaussian random variables with unit variance at each discrete time instant.

In the correlated fading case, the path gains were generated according to the statistical model described in [74]. The base station (BS) was the transmitter and the mobile terminal (MT) was the receiver. Both the BS and the MT were assumed to have a uniform, linear array of isotropic antennas, and the MT was surrounded by a ring of scatterers. The model parameters were: d_B - BS antenna separation, d_M - MT antenna separation, D - distance between the BS and the MT, R - radius of the scatterer ring, N_s - number of scatterers, β - direction of the BS antenna

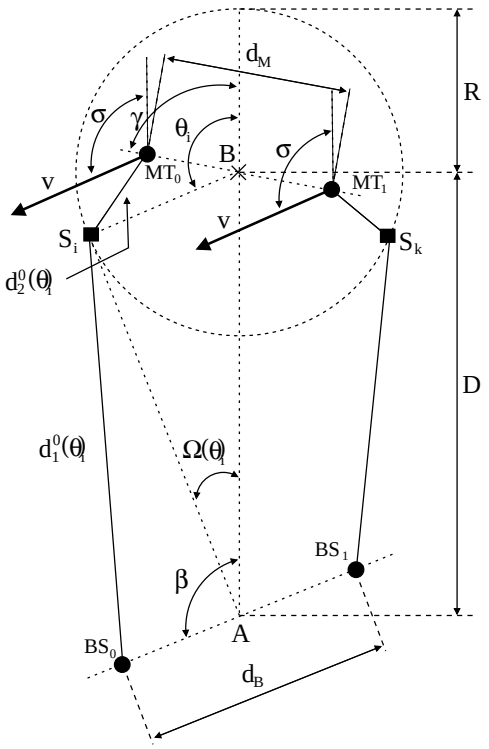

Figure 3.2: Geometric model for correlated fading simulations

array, γ - direction of the MT antenna array, σ - direction of the MT movement, v - the magnitude of the MS speed, f_c - the carrier frequency (or λ_c - the carrier wavelength), and T_s - the channel symbol period.

The geometry of the model is shown in Figure 3.2 for two adjacent BS antennas (BS_0, BS_1) and two adjacent MT antennas (MT_0, MT_1). The ith ($i = 0, 1, \ldots, N_s - 1$) scatterer was at an angle θ_i from the middle point of the MT antenna array. For each frame, the scatterer angles were randomly generated in the range $[-\pi, \pi]$ with uniform distribution. The effect of scatterer i was modeled as multiplication of the incident signal by a scattering coefficient S_i. The scattering coefficients were modeled as independent, complex, zero mean, Gaussian random variables with variance $1/N_s$. Assuming that the arrival time differences between the signals coming from different scatterers are negligible compared to one channel

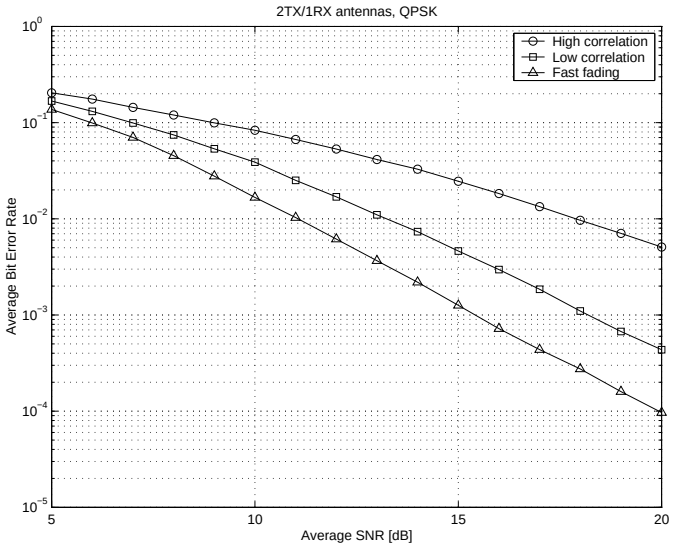

Figure 3.3: ST code for 2 antennas, QPSK

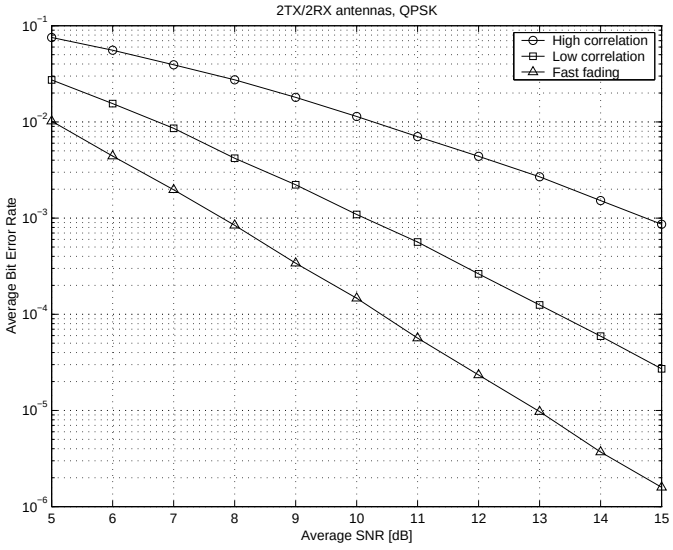

Figure 3.4: ST code for 2 antennas, QPSK

symbol period (i.e. flat fading), the path gain between transmit antenna k and receive antenna l at discrete time t ($t = 0, 1, \ldots$) was calculated as

$$\alpha_{k,l}(t) = \sum_{i=0}^{N_s-1} S_i e^{j2\pi f_D \cos(\theta_i - \sigma) T_s t} e^{-j\frac{2\pi}{\lambda_c}(d_1^k(\theta_i) + d_2^l(\theta_i))}, \tag{3.32}$$

where f_D ($f_D = v/\lambda_c$) is the maximum Doppler shift, $d_1^k(\theta_i)$ is the distance between transmit antenna k and scatterer i, and $d_2^l(\theta_i)$ is the distance between scatterer i and receive antenna l. It can be easily seen that for each k, l and t, $\alpha_{k,l}(t)$ is a complex, zero mean, Gaussian random variable with unit variance.

During the simulations, we used the following parameter values: $D = 1$km, $R = 20$m, $N_s = 20$, $\beta = \frac{3\pi}{4}$ rad, $\gamma = \frac{\pi}{4}$ rad, $\sigma = \frac{3\pi}{4}$ rad, and $v = 70$km/h. Three cases were considered: (a) high correlation ($T_s = 50\mu$s, $f_c = 900$MHz, $d_B = 5\lambda_c$, $d_M = 0.6\lambda_c$), (b) low correlation ($T_s = 500\mu$s, $f_c = 2$GHz, $d_B = 25\lambda_c$, $d_M = 5\lambda_c$), and (c) no correlation (fast fading). Note that the value of d_M is significant only if the MT has multiple receive antennas.

We present probability of bit error curves as functions of the average SNR per source symbol at the receive antennas. Figure 3.3 depicts the performance of the ST code designed by our method for 2 transmit antennas and QPSK constellation ($K = 2$, $B = 4$, $N = 4$) with 1 receive antenna. The bit error rate curves for the same code with 2 receive antennas are shown in Figure 3.4. Both curves demonstrate that the spatio-temporal correlation has a significant impact on the performance. Moreover, it can be observed that in the low correlation case, the bit error probability curve becomes approximately parallel to the fast fading bit error probability curve at high SNR. Therefore, they achieve the same diversity level, validating our analysis.

Since we are not aware of any other code construction method for space-time correlated channels, we compare our method with the ST codes of [66], [67] de-

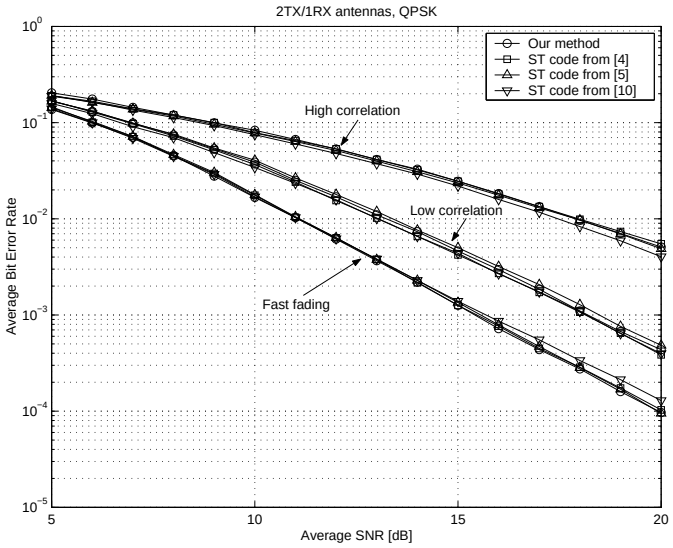

Figure 3.5: ST codes for 2 antennas, QPSK

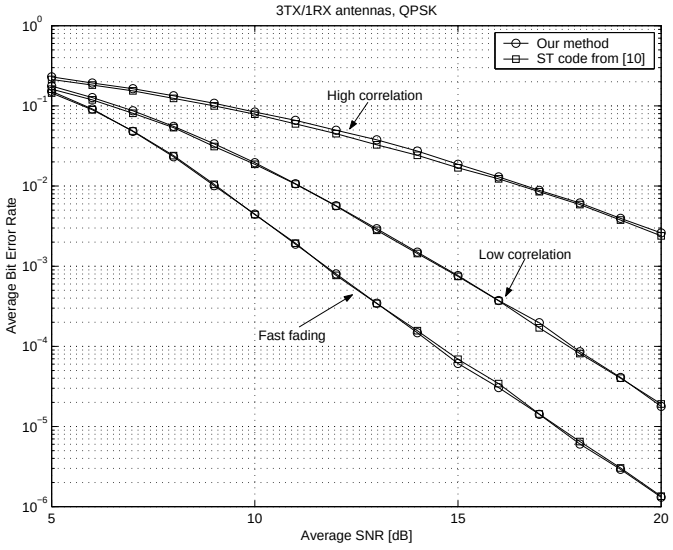

Figure 3.6: ST codes for 3 antennas, QPSK

τ	2	3	4	5	6
δ_τ	24	32	64	128	256
δ_τ [63]	16	32	96	128	256

Table 3.2: Minimum norm products for 2 antennas, QPSK

signed for fast fading channels, and the ST code of [63] designed for quasi-static fading channels. These ST codes represent point-solutions in the design space, and they are used to verify the performance of the codes obtained via the proposed systematic construction method in the special cases where previously known solutions exist.

Figure 3.5 shows the performance of the ST codes constructed for a 2 transmit antenna system and QPSK modulation ($K = 2$, $B = 4$, $N = 4$) with 1 receive antenna. All of these codes satisfy the uniqueness criterion. It is observed that all codes have essentially the same performance, with the ST code from [63] being a little better in the high correlation case and being a little worse in the fast fading case. Note that our systematic design method resulted in a ST code that achieves the same performance as the ST code of [67], which was found by exhaustive search. The theoretical performance of the above codes was also compared. Table 3.2 shows the minimum norm product values δ_τ for the τ-long error events. The entries in the second row of the table correspond to the ST codes of [66], [67] and our method, while the third row contains the δ_τ values for the ST code of [63]. The table entries were obtained by performing computer search. These δ_τ values predict the similar performance of the methods described in [66], [67] and the proposed approach, along with the slightly worse performance of the ST code of [63] at high SNR.

The bit error rate curves for 3 transmit antennas and QPSK modulation ($K = 3$, $B = 4$, $N = 16$) with 1 receive antenna are depicted in Figure 3.6. Our code (the

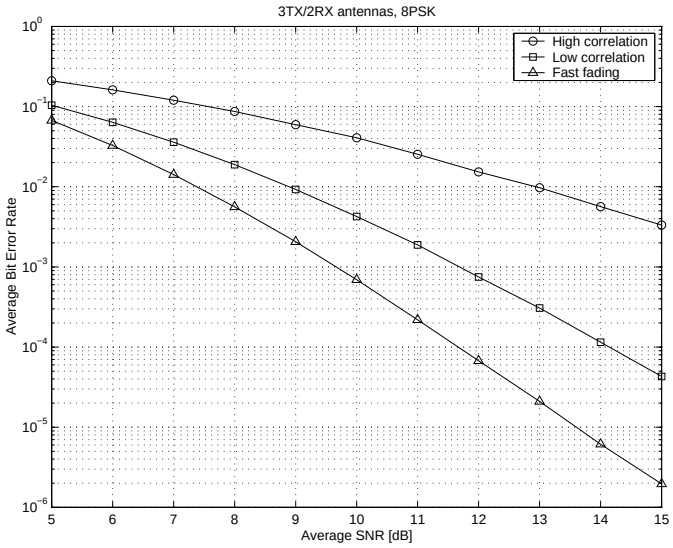

Figure 3.7: ST codes for 3 antennas, 8PSK

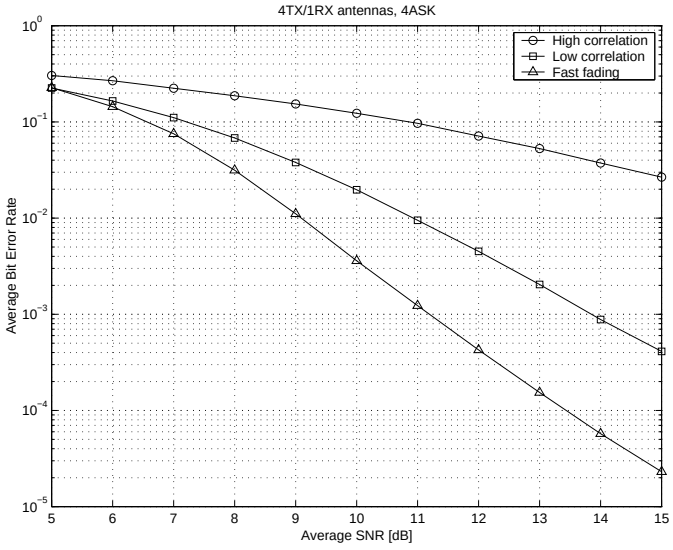

Figure 3.8: ST codes for 4 antennas, 4ASK

τ	3	4	5	6
δ_τ	384	384	2304	6144
δ_τ [63]	216	576	1728	2880

Table 3.3: Minimum norm products for 3 antennas, QPSK

ST code shown in Figure 3.1) is compared to the ST code described in [63]. These codes also satisfy the uniqueness criterion. The performance of the two codes is almost identical, and the bit error curves for the fast fading channel model and for the low correlation channel model are approximately parallel at high SNR. The ST code of [63], also found by computer search, performs slightly better in the high correlation case, which is expected since this code was designed for quasi-static channels. The minimum norm product values for the proposed method are given in the second row of Table 3.3, and the third row contains the values for the ST code given in [63]. The theoretical norm product values confirm the tendencies observed in Figure 3.6.

The performance of our ST code constructed for 3 transmit antennas and 8PSK modulation ($K = 3$, $B = 8$, $N = 64$) with 2 receive antennas is shown in Figure 3.7. The bit error rate curves of our 4 antenna 4ASK ST code ($K = 4$, $B = 4$, $N = 64$) with 1 receive antenna can be observed in Figure 3.8. The behavior of the curves are similar to that of the 2 and 3 transmit antenna cases. To the best of our knowledge, no ST codes have been published that we could compare these codes against.

3.8 Chapter Summary

We derived the performance criteria for space-time correlated flat Rayleigh fading channels, and we developed a systematic ST trellis code design method for an arbitrary number of transmit antennas and any memoryless modulation. Based on the theoretical and experimental results, we can draw the following conclusion.

If the space-time channel is not heavily correlated (i.e. the space-time correlation matrix is of full rank), the space-time code design problem for correlated channels can be reduced to the code design problem for fast fading channels, and it is possible to achieve the diversity level available in fast fading environment. Moreover, for communication systems having K transmit and L receive antennas, the space-time correlated channel model does not limit the maximum achievable diversity level to KL, as in the case of the quasi-static channel model. By increasing the number of encoder and decoder states (increasing the length of the shortest error event), arbitrarily high diversity order can be achieved.

The ST trellis codes constructed by the proposed method were also compared to existing ST codes that represent point-solutions in our design space. The simulations showed that our systematic design procedure results in codes that have the same performance as the codes previously found by computer search, so our codes are expected to perform very well in cases where exhaustive computer search is not feasible. We also constructed ST trellis codes for 3 and 4 transmit antennas and 8PSK and 4ASK modulations. For these design parameters and non-quasi-static channel models, no other ST trellis codes exist in the literature.

Chapter 4

Fast Decoding of Space-Frequency Block Codes

4.1 Introduction

The recently proposed space-time (ST) and space-frequency (SF) coded multiple-input-multiple-output (MIMO) systems have promised considerable performance improvement over the single-antenna systems. However, the computational complexity of the maximum likelihood (ML) decoding algorithm may hamper the widespread use of such systems, so the development of low complexity ST and SF decoding algorithms is a problem of paramount importance.

Computationally efficient decoding algorithms have only been proposed for decoding ST block codes in quasi-static, flat fading environment [32], [33], [76]. For ST block codes transmitted over temporally evolving channels and for SF block-coded MIMO-OFDM systems, where the channel changes along the frequency axis, low complexity decoding algorithms still do not exist in the literature.

The sphere decoding algorithm was introduced in [75] assuming a single-antenna,

real-valued fading channel model. Later results [76] [77] [79], generalized the algorithm to complex valued MIMO systems. In [80], the sphere decoding algorithm was applied to equalize uncoded MIMO systems. All of these works considered the transmission of uncoded channel symbols (i.e no dependence among the channel symbols transmitted at different times from different transmit antennas), or ST coded systems that can be transformed into equivalent uncoded systems, assumed that the channel is quasi-static and formulated the sphere decoding problem in the real domain, so the proposed algorithms can only be used with modulation methods that can be decomposed into real and imaginary parts (for example, square QAM).

In this chapter, we propose a computationally efficient decoding algorithm for SF block codes. We formulate the sphere decoding problem in the complex domain, which allows us to fully exploit the distance structure of complex signal constellations. We develop a systematic method to transform the decoding problem into an equivalent representation that is more appropriate for the purpose of sphere decoding. Then, we propose a modulation independent sphere decoding framework by interpreting the sphere decoding problem as a greedy, constrained depth-first search. Due to the modular structure of the framework, it can be used to construct a decoding algorithm that can be used with any memoryless modulation, and it can also be tailored to a particular modulation method by taking full advantage of the geometric properties of the chosen signal constellation. Finally, for square QAM and PSK constellations, we propose a fast, nearest neighbor search algorithm that can considerably reduce the decoding complexity.

The organization of this chapter is as follows. Section 4.2 will introduce the system model and the used notation. Section 4.3 will provide the problem formu-

lation. An overview of the sphere decoding approach will be provided in Section 4.4, and this section will also summarize the state of the art. Section 4.5 will describe the proposed algorithm, and Section 4.6 will show and discuss the simulation results, and Section 4.7 will summarize the content of this chapter.

4.2 System Model and Notation

Consider a SF-coded MIMO-OFDM system having K transmit antennas, L receive antennas and M sub-carriers, with M being a multiple of K. Suppose that the frequency selective fading channels between each pair of transmit and receive antennas have P independent delay paths and the same power delay profile. The MIMO channel is assumed to be constant over each OFDM block period. The channel impulse response from transmit antenna k to receive antenna l at time τ is modeled as

$$h_{k,l}(\tau) = \sum_{p=0}^{P-1} \alpha_{k,l}(p)\delta(\tau - \tau_p), \tag{4.1}$$

where τ_p is the delay and $\alpha_{k,l}(p)$ is the complex amplitude of the p-th path between transmit antenna k and receive antenna l. The $\alpha_{k,l}(p)$'s are modeled as zero-mean, complex Gaussian random variables with variances $E[|\alpha_{k,l}(p)|^2] = \delta_p^2$. The powers of the P paths are normalized such that $\sum_{p=0}^{P-1} \delta_p^2 = 1$. From (4.1), the frequency response of the channel is given by

$$H_{k,l}(f) = \sum_{p=0}^{P-1} \alpha_{k,l}(p)e^{-j2\pi f\tau_p}. \tag{4.2}$$

We assume that the MIMO channel is spatially uncorrelated, i.e. the channel taps $\alpha_{k,l}(p)$ are independent for different indices (k, l).

The input bit stream is divided into b bit long segments, creating B-ary ($B = 2^b$) source symbols. The encoder forms M/K source symbol blocks, each containing

n_s source symbols. Source symbol $s_i \in \{0, 1, \ldots, B-1\}, i = 0, 1, \ldots, n_s - 1.$ is mapped onto a complex channel symbol (or constellation point) x_i according to

$$x_i = \Omega(s_i), \quad i = 0, 1, \ldots, n_s - 1,$$

where the function $\Omega(.)$ represents the modulation operation. The average energy of the constellation will be denoted by E_{avg}.

Then, the SF encoder forms two-dimensional, square codewords from the channel symbols. The SF codeword corresponding to the t-th $(t = 0, 1, \ldots, M/K - 1)$ source symbol block can be expressed as a K by K matrix $\mathbf{C}$:

$$\mathbf{C} = \begin{bmatrix} c_1[Kt] & c_2[Kt] & \cdots & c_{K-1}[Kt] \\ c_1[Kt+1] & c_2[Kt+1] & \cdots & c_{K-1}[Kt+1] \\ \vdots & \vdots & \ddots & \vdots \\ c_1[Kt+K-1] & c_2[Kt+K-1] & \cdots & c_{K-1}[Kt+K-1] \end{bmatrix}, \tag{4.3}$$

where $c_k[i]$ denotes the channel symbol transmitted over the i-th sub-carrier by transmit antenna k. We assume that each $c_k[i]$ is either zero, or a channel symbol or a negative and/or complex conjugate of a channel symbol corresponding to a source symbol in the appropriate source symbol block.

At the receiver, after matched filtering, removing the cyclic prefix, and applying FFT, the received signal corresponding to the t-th source symbol block at sub-carrier $Kt + m$ $(m = 0, 1, \ldots, K-1)$ and receive antenna l is given by

$$y_l[Kt+m] = \sum_{k=0}^{K-1} H_{k,l}[Kt+m] c_k[Kt+m] + n_l[Kt+m], \tag{4.4}$$

where $H_{k,l}[i] = H_{k,l}(i\Delta f)$ is the channel frequency response at the i-th sub-carrier between transmit antenna k and receive antenna l, $\Delta f = 1/T$ is the sub-carrier separation in the frequency domain, and T is the OFDM symbol period. We assume that the channel state information $H_{k,l}[i]$ is known at the receiver, but not at the transmitter. In (4.4), $n_l[i]$ denotes the additive complex Gaussian noise at

the i-th sub-carrier at receive antenna l. The noise samples $n_l[i]$ are assumed to be independent for different l's and i's.

In the sequel, we will focus our attention on decoding a single block, so a simplified notation will be used by dropping the block index:

$$y_l[m] = \sum_{k=0}^{K-1} H_{k,l}[m]c_k[m] + n_l[m], \tag{4.5}$$

for $m = 0, 1, \ldots, K-1$. The noise samples $n_l[m]$ are assumed to have zero mean and variance $1/(\rho\gamma)$, where the scaling factor γ is defined as

$$\gamma = \frac{b}{K^2}\frac{1}{E_{avg}},$$

so ρ will be the signal to noise ratio per bit at each sub-carrier at each receive antenna.

The above formulation can also be used to decode space-time block codes transmitted over quasi-static or temporally evolving flat fading channels. In that case, the frequency domain (the dimension along the index m) needs to be replaced by the discrete time domain, and $H_{k,l}[m]$ will be the path gain between transmit antenna k and receive antenna l at time m.

In the sequel, the notation $||.||$ will stand for the Frobenius norm of a vector or a matrix. The functions Re{.} and Im{.} extract the real and imaginary parts,respectively, of a complex quantity.

4.3 Problem Formulation

In general, SF coding introduces spatial and frequency-domain dependence among the code symbols $c_k[i]$ within a code block $\mathbf{C}$. For example, in case of the 2×2 orthogonal design [32]

$$\mathbf{C} = \begin{bmatrix} x_0 & x_1 \\ -x_1^* & x_0^* \end{bmatrix}, \tag{4.6}$$

the channel symbols transmitted from different transmit antennas and through different sub-carriers are clearly related. However, to be able to use a sphere decoder (to be able to make sequential decisions on the sent signal coordinates), it is necessary to transform the received signal to an equivalent signal representation, where the coordinates of the sent signal vector are independent. We have chosen orthogonal and block-orthogonal designs of square size to construct SF codes since in case of those codes, the transformation can be carried out easily. Orthogonal and block-orthogonal designs can ensure that the code difference matrix corresponding to any two distinct codewords have full rank [33], [41], achieving full spatial diversity (a diversity order of KL). For 2, 3 and 4 transmit antennas, they can also provide full symbol rate (1 symbol per sub-channel per channel use). They cannot achieve full spatial and frequency diversity (a diversity order of KLP) [51]. However, full-diversity SF codes can be constructed from orthogonal designs via a mapping [56] [57], and this issue will be considered later in this section.

4.3.1 Two Transmit Antenna Case

For 2 transmit antennas ($K = 2$), the 2×2 orthogonal design (4.6) with symbol rate 1 ($n_s = K = 2$). is used to construct SF codes. From (4.5), the received signal at receive antenna l can be expressed as

$$\begin{aligned} y_l[0] &= H_{0,l}[0]x_0 + H_{1,l}[0]x_1 + n_l[0], \text{ and} \\ y_l[1] &= -H_{0,l}[1]x_1^* + H_{1,l}[1]x_0^* + n_l[1]. \end{aligned} \tag{4.7}$$

By taking the complex conjugate of the second line of (4.7), we obtain

$$y_l^*[1] = -H_{0,l}^*[1]x_1 + H_{1,l}^*[1]x_0 + n_l^*[1],$$

so the transformed equivalent received signal vector $\mathbf{y}_l = [\, y_l[0], y_l^*[1] \,]^T$ for receive antenna l can be rewritten in matrix-vector form as

$$\mathbf{y}_l = \mathbf{H}_l \mathbf{x} + \mathbf{n}_l,$$

where $\mathbf{x} = [\, x_0, x_1 \,]^T$ is the transmitted channel symbol vector, $\mathbf{n}_l = [\, n_l[0], n_l^*[1] \,]^T$ is the equivalent noise component, and $\mathbf{H}_l$ is defined as

$$\mathbf{H}_l = \begin{bmatrix} H_{0,l}[0] & H_{1,l}[0] \\ H_{1,l}^*[1] & -H_{0,l}^*[1] \end{bmatrix}.$$

By collecting the received signal and noise components corresponding to different receive antennas in $KL \times 1$ vectors as $\mathbf{y} = [\mathbf{y}_0^T, \ldots, \mathbf{y}_{L-1}^T]^T$, and $\mathbf{n} = [\mathbf{n}_0^T, \ldots, \mathbf{n}_{L-1}^T]^T$, the equivalent received signal can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{4.8}$$

where the $KL \times K$ matrix $\mathbf{H}$ is the equivalent channel matrix, defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 \\ \vdots \\ \mathbf{H}_{L-1} \end{bmatrix}. \tag{4.9}$$

Note that the above described equivalent representation has the following properties that are important from the viewpoint of the sphere decoding algorithm. First, the coordinates of the noise vector $\mathbf{n}$ are independent, zero mean, complex Gaussian random variables with variance $1/(\rho\gamma)$. Second, the coordinates of the $\mathbf{x}$ vector are independent. Third, the matrix $\mathbf{H}$ has at least as many rows as columns, independently of the number of receive antennas. Fourth, since the entries in the matrix $\mathbf{H}$ are complex, zero mean, Gaussian random variables and we have assumed that the MIMO channel is spatially independent, the matrix $\mathbf{H}$ has full (column) rank with high probability.

4.3.2 Four Transmit Antenna Case

In case of 4 transmit antennas ($K = 4$), we have adopted the full-rank block orthogonal design [41]

$$\mathbf{C} = \begin{bmatrix} x_0 & x_1 & x_2 e^{j\phi} & x_3 e^{j\phi} \\ -x_1^* & x_0^* & -x_3^* e^{-j\phi} & x_2^* e^{-j\phi} \\ x_2 e^{j\phi} & x_3 e^{j\phi} & x_0 & x_1 \\ -x_3^* e^{-j\phi} & x_2^* e^{-j\phi} & -x_1^* & x_0^* \end{bmatrix}, \tag{4.10}$$

which provides a symbol rate 1 ($n_s = K = 4$). In (4.10), the channel symbols x_i are taken from the same constellation, and the rotation angle ϕ is chosen in such a way that it can ensure the full rank of the code difference matrix for any two distinct code matrices. The value of the optimal angle that maximizes the coding advantage depends on the used constellation. Proceeding similarly to Section 4.3.1, the equivalent received signal vector for receive antenna l can be obtained as $\mathbf{y}_l = [\, y_l[0], y_l^*[1], y_l[2], y_l^*[3] \,]^T$, and the equivalent noise vector becomes $\mathbf{n}_l = [\, n_l[0], n_l^*[1], n_l[2], n_l^*[3] \,]^T$. The result is the matrix equation $\mathbf{y}_l = \mathbf{H}_l \mathbf{x} + \mathbf{n}_l$ with $\mathbf{x} = [\, x_0, x_1, x_2, x_3 \,]^T$ and the matrix $\mathbf{H}_l$, defined as

$$\mathbf{H}_l = \begin{bmatrix} H_{0,l}[0] & H_{1,l}[0] & H_{2,l}[0]e^{j\phi} & H_{3,l}[0]e^{j\phi} \\ H_{1,l}^*[1] & -H_{0,l}^*[1] & H_{3,l}^*[1]e^{j\phi} & -H_{2,l}^*[1]e^{j\phi} \\ H_{2,l}[2] & H_{3,l}[2] & H_{0,l}[2]e^{j\phi} & H_{1,l}[2]e^{j\phi} \\ H_{3,l}^*[3] & -H_{2,l}^*[3] & H_{1,l}^*[3]e^{j\phi} & -H_{0,l}^*[3]e^{j\phi} \end{bmatrix}.$$

The $\mathbf{y}$ and $\mathbf{n}$ vectors are formed similarly to the 2 transmit antenna case, and the equivalent received signal is given by (4.8), with the $KL \times K$ equivalent channel matrix $\mathbf{H}$ formatted according to (4.9). All the properties of $\mathbf{H}$ described in Section 4.3.1 also hold. Note that to make the decoding process easier, the effect of the constellation rotation has been included in the equivalent channel matrix.

The full-rate SF code that achieves full spatial diversity for 3 transmit antennas ($K = 3$) can be easily obtained from the 4×4 block-orthogonal design by deleting one column from the code matrix $\mathbf{C}$, yielding a SF code with symbol rate 1 ($n_s = 4$).

The resulting equivalent channel model is very similar to the 4 transmit antenna case, the only difference is that the entries in the matrices $\mathbf{H}_l$ corresponding to the deleted transmit antenna index will be zero. For example, if the last column of $\mathbf{C}$ is deleted (antenna index three), $\mathbf{H}_l$ becomes

$$\mathbf{H}_l = \begin{bmatrix} H_{0,l}[0] & H_{1,l}[0] & H_{2,l}[0]e^{j\phi} & 0 \\ H^*_{1,l}[1] & -H^*_{0,l}[1] & 0 & -H^*_{2,l}[1]e^{j\phi} \\ H_{2,l}[2] & 0 & H_{0,l}[2]e^{j\phi} & H_{1,l}[2]e^{j\phi} \\ 0 & -H^*_{2,l}[3] & H^*_{1,l}[3]e^{j\phi} & -H^*_{0,l}[3]e^{j\phi} \end{bmatrix}.$$

4.3.3 Full-Diversity SF Codes

To our knowledge, full-rate (symbol rate 1) and full-diversity (diversity order of KLP) SF codes have not been proposed yet. In [56] [57], a method was proposed to construct full-diversity SF codes from full-rank ST codes via a repetition mapping, trading off data rate for performance. It was shown that by repeating each row of the ST code matrix p times ($p \leq P$), a diversity order of KLp can be achieved. For instance, for MIMO channels with at least two independent delay paths ($P \geq 2$), the SF code given in (4.6) achieves a diversity of $2L$, while the repetition coded SF code

$$\mathbf{C} = \begin{bmatrix} x_0 & x_1 \\ x_0 & x_1 \\ -x_1^* & x_0^* \\ -x_1^* & x_0^* \end{bmatrix} \tag{4.11}$$

can achieve a diversity order of $4L$. Now we demonstrate that this construction can also be transformed into an equivalent representation that is convenient for sphere decoding through a simple $K = 2$, $L = 1$ example with repetition 2. In this case, the equivalent received signal vector becomes $\mathbf{y} = [\, y_0[0], y_0[1], y_0^*[2], y_0^*[3] \,]^T$, and the equivalent noise component is $\mathbf{n} = [\, n_0[0], n_0[1], n_0^*[2], n_0^*[3] \,]^T$. Then, the equivalent received signal vector can be expressed as in (4.8), with the channel

symbol vector $\mathbf{x} = [\, x_0, x_1 \,]^T$ and the equivalent channel matrix

$$\mathbf{H} = \begin{bmatrix} H_{0,0}[0] & H_{1,0}[0] \\ H_{0,0}[1] & H_{1,0}[1] \\ H^*_{1,0}[2] & -H^*_{0,0}[2] \\ H^*_{1,0}[3] & -H^*_{0,0}[3] \end{bmatrix}.$$

The generalization for 3 and 4 transmit antennas, more receive antennas and repetitions is straightforward.

4.4 Sphere Decoding

From communication perspective, the sphere decoding problem can be formulated as follows. Assume that for a communication system, the received $M \times 1$ signal vector $\mathbf{y}$ can be expressed as

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

where $\mathbf{x}$ is the $M \times 1$ sent complex signal vector taken from a finite set $\mathcal{X}$, $\mathbf{H}$ is the $M \times N$ complex channel matrix, assumed to be known by the receiver, and $\mathbf{n}$ is the $M \times 1$ zero mean, white, complex Gaussian receiver noise vector. We assume that $M \geq N$, the coordinates of $\mathbf{x}$ are independent and $\mathbf{H}$ is of full column rank (rank N). In case of the SF-coded MIMO-OFDM systems, we have $M = KL$ and $N = n_s$.

The maximum likelihood decoding amounts to finding the vector $\hat{\mathbf{x}}_{ML}$ such that

$$\hat{\mathbf{x}}_{ML} = \arg \min_{\mathbf{x} \in \mathcal{X}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||^2.$$

Unfortunately, in some cases, this can only be performed by exhaustive search over the elements of $\mathcal{X}$, which may result in prohibitively high computational complexity. To alleviate this computational burden, sphere decoding was proposed [75] [76] [77] [78] [79] [80], where the decoder tries to identify a subset of $\mathcal{X}$ that

results in noiseless received signal vectors that lie within a hyper-sphere of radius r centered around the the actually received (noisy) signal vector. Then, the decoder only searches over this subset to find the signal vector that is closest to the center point. More formally:

1) Find all vectors $\mathbf{x} \in \mathcal{X}$ satisfying $||\mathbf{y} - \mathbf{H}\mathbf{x}||^2 \leq r^2$.

2) Declare the code vector minimizing $||\mathbf{y} - \mathbf{H}\mathbf{x}||^2$ to be the decoded code vector.

The basic idea behind the method is to transform the quantity $||\mathbf{y} - \mathbf{H}\mathbf{x}||^2$ in such a form that the decisions on the coordinates of $\mathbf{x}$ can be made sequentially. An implementation of the algorithm [79] [80] proceeds in the following way. First, the problem is expressed in the real domain:

$$\mathbf{y}_R = \mathbf{H}_R\mathbf{x}_R + \mathbf{n}_R,$$

where $\mathbf{y}_R = [\mathrm{Re}\{\mathbf{y}^T\}, \mathrm{Im}\{\mathbf{y}^T\}]^T$, $\mathbf{x}_R = [\mathrm{Re}\{\mathbf{x}^T\}, \mathrm{Im}\{\mathbf{x}^T\}]^T$, $\mathbf{n}_R = [\mathrm{Re}\{\mathbf{n}^T\}, \mathrm{Im}\{\mathbf{n}^T\}]^T$, and the $2M \times 2N$ real matrix $\mathbf{H}_R$ is defined as

$$\mathbf{H}_R = \begin{bmatrix} \mathrm{Re}\{\mathbf{H}\} & -\mathrm{Im}\{\mathbf{H}\} \\ \mathrm{Im}\{\mathbf{H}\} & \mathrm{Re}\{\mathbf{H}\} \end{bmatrix}.$$

It is assumed that the coordinates of $\mathbf{x}_R$ are integers (i.e. the set $\mathcal{X}$ contains points from a 2-dimensional square lattice). Then, the the zero-forcing solution $\mathbf{z}_R = \mathbf{H}_R^+\mathbf{y}_R$ is found, where $\mathbf{H}_R^+$ is the pseudo-inverse of $\mathbf{H}_R$, defined as $\mathbf{H}_R^+ = (\mathbf{H}_R^T\mathbf{H}_R)^{-1}\mathbf{H}_R^T$. The vector $\mathbf{z}_R$ can also be thought of as a vector that contains the coordinates of the received signal point $\mathbf{y}_R$ with respect to the lattice generated by $\mathbf{H}_R$. Since $||\mathbf{y} - \mathbf{H}\mathbf{x}||^2 = ||\mathbf{y}_R - \mathbf{H}_R\mathbf{x}_R||^2$ can be rewritten as [80]

$$||\mathbf{y}_R - \mathbf{H}_R\mathbf{x}_R||^2 = ||\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R)||^2 + ||\mathbf{y}_R||^2 - ||\mathbf{H}_R\mathbf{z}_R||^2,$$

the sphere decoding problem can be expressed in a different form: Find all vectors $\mathbf{x}_R$ that satisfy $||\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R)||^2 \leq r'^2$ with $r'^2 = r^2 - ||\mathbf{y}_R||^2 + ||\mathbf{H}_R\mathbf{z}_R||^2$. The next step is to calculate the real Cholesky (or QR) decomposition of the matrix $\mathbf{H}_R^T\mathbf{H}_R$, yielding a $2N \times 2N$ upper triangular matrix $\mathbf{R}_R$ with positive diagonal entries such that $\mathbf{R}_R^T\mathbf{R}_R = \mathbf{H}_R^T\mathbf{H}_R$. By taking advantage of the relationship

$$\begin{aligned} ||\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R)||^2 &= (\mathbf{z}_R - \mathbf{x}_R)^T\mathbf{H}_R^T\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R) = \\ &= (\mathbf{z}_R - \mathbf{x}_R)^T\mathbf{R}_R^T\mathbf{R}_R(\mathbf{z}_R - \mathbf{x}_R) = ||\mathbf{R}_R(\mathbf{z}_R - \mathbf{x}_R)||^2, \end{aligned}$$

and obtaining a $2N \times 2N$ matrix $\mathbf{Q}_R = \{Q_{k,l}\}$ from the entries of the matrix $\mathbf{R}_R = \{R_{k,l}\}$ as

$$Q_{k,l} = \begin{cases} R_{k,k}^2 & \text{if } k = l \\ \frac{R_{k,l}}{R_{k,k}} & \text{otherwise,} \end{cases} \tag{4.12}$$

$||\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R)||^2$ can be expressed as [75]

$$||\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R)||^2 = \sum_{k=0}^{2N-1} Q_{k,k}\left(z_k - x_k + \sum_{l=k+1}^{2N-1} Q_{k,l}(z_l - x_l)\right)^2. \tag{4.13}$$

In (4.13), x_k and z_k ($k = 0, 1, \ldots, 2N-1$) denote the k-th coordinates of $\mathbf{x}_R$ and $\mathbf{z}_R$, respectively. The next step is to define the quantities T_k and α_k ($k = 0, 1, \ldots, 2N-1$) recursively as follows:

$$T_k = \begin{cases} r'^2 & \text{for } k = 2N-1 \\ T_{k+1} - Q_{k+1,k+1}(\alpha_{k+1} - x_{k+1})^2 & \text{for } k = 2N-2, 2N-3, \ldots, 0, \end{cases} \tag{4.14}$$

and

$$\alpha_k = z_k + \sum_{l=k+1}^{2N-1} Q_{k,l}(z_l - x_l). \tag{4.15}$$

The quantity T_k can be thought of as the modified partial sphere constraint for x_k given $x_{2N-1}, x_{2N-2}, \ldots, x_{k+1}$, and α_k can be interpreted as the k-th received

signal component given $x_{2N-1}, x_{2N-2}, \ldots, x_{k+1}$. The algorithm that finds all possible values of x_k satisfying the sphere constraint assuming that T_{k+1}, α_{k+1} and $x_{2N-1}, x_{2N-2}, \ldots, x_{k+1}$ have already been determined can be described as [75]:

1) Calculate T_k and α_k according to (4.14) and (4.15).

2) Find all values of x_k such that $(\alpha_k - x_k)^2 \leq T_k/Q_{k,k}$. Since x_k are assumed to be integers, the range for the possible values of x_k will be constrained as $L_k \leq x_k \leq U_k$, where

$$L_k = \left\lceil \alpha_k - \sqrt{\frac{T_k}{Q_{k,k}}} \right\rceil, \quad \text{and} \quad U_k = \left\lfloor \alpha_k + \sqrt{\frac{T_k}{Q_{k,k}}} \right\rfloor.$$

The size of the used constellation (i.e. the maximum and minimum values of x_k) will also limit the range. Now, starting form the lower bound L_k and finishing at the upper bound U_k, it is easy to enumerate only those x_k values that satisfy the current partial constraint.

Therefore, to find all $\mathbf{x}_R$ vectors satisfying the sphere constraint, we start with the last coordinate, x_{2N-1}, and enumerate all admissible values it can take on using steps 1) and 2). Then, we move one level further, and for each x_{2N-1} value, we go over all possible values of x_{2N-2} in a recursive manner. By repeating the same procedure, we reach the last level, making decisions on x_0.

Once a valid $\mathbf{x}_R$ vector is found, the sphere radius can be adjusted so that the point $\mathbf{H}_R\mathbf{x}_R$ would lie exactly on the surface of the new sphere:

$$r'^2 = T_{2N-1} - T_0 + Q_{0,0}(\alpha_0 - x_0)^2.$$

Then, the search continues from x_{2N-1} again, making sure that each $\mathbf{x}_R$ vector is only considered once. By shrinking the sphere, further complexity reduction can be achieved, as the algorithm excludes some of the $\mathbf{x}_R$ vectors that would generate

$\mathbf{H}_R\mathbf{x}_R$ vectors lying further away from the center than the signal point found most recently. Since we are looking for the closest signal point, this step does not result in any performance degradation.

The algorithm proposed in [79] suggested further improvement. It is based on the observation that if the algorithm can find a signal vector $\mathbf{x}_R$ such that the corresponding $\mathbf{H}_R\mathbf{x}_R$ vector is very close to the center $\mathbf{y}_R$, most of the candidate $\mathbf{x}_R$ vectors can be eliminated without enumerating them by decreasing the sphere radius. Therefore, the algorithm at level k enumerates all possible values of x_k, sorts them according to the metric $d_k = (\alpha_k - x_k)^2$ and explores the search space in this order (the x_k closest to α_k is searched first). If a valid $\mathbf{x}_R$ vector is found, the partial constraints T_k and the bounds L_k and U_k for $k = 2N-1, 2N-2, \ldots, 0$ are reevaluated, and at all levels, the candidate solutions not satisfying the new partial constraints can be easily eliminated. When the algorithm terminates, the most recently found $\mathbf{x}_R$ vector is declared as the decoded signal vector. This approach cannot guarantee that the first $\mathbf{x}_R$ that is found is the closest signal point as it makes greedy decisions at each level, but most of the time it can significantly reduce the search space.

Note that the algorithm in [79] assumed that $||\mathbf{y}_R - \mathbf{H}_R\mathbf{x}_R||^2 = ||\mathbf{H}_R(\mathbf{z}_R - \mathbf{x}_R)||^2$, so it can only be used in the $M = N$ case. On the other hand, the authors of [80] fixed the radius of the sphere, so the algorithm presented here is actually a combination of those in [79] and [80].

The decoding problem for the $M < N$ case was considered in [77], [78] for full row-rank channel matrices. By calculating the general Cholesky factorization of the matrix $\mathbf{H}_R^T\mathbf{H}_R$, a sphere decoding algorithm similar to the one described above can be formulated. However, the resulting $\mathbf{R}_R$ matrix will have zero entries $R_{k,l} = 0$

for $k = 2M, 2M+1, \ldots, 2N-1$, so the last $2(N-M)$ diagonal entries of the $\mathbf{Q}$ matrix will also be zero. (The $Q_{k,l}$, $k \neq l$, values for $k = 2M, 2M+1, \ldots, 2N-1$ are by definition zero.) As a consequence, for $k = 2M, 2M+1, \ldots, 2N-1$, the lower bounds L_k become $-\infty$ and the upper bounds U_k become $+\infty$, requiring exhaustive search along those signal coordinates. This means that if N is considerably larger than M, the decoding complexity may become impractically high, so it is more desirable to formulate the problem in such a way that the resulting dimensions would yield $M \geq N$.

4.5 The Proposed Algorithm

Despite the recent advances in sphere decoding described in the previous section, the existing methods have some disadvantages. Most decoding approaches were formulated in the real domain, limiting their use in cases where the applied complex constellation can be decomposed into the product of two real constellations (for example, square QAM). Moreover, as a result of real domain processing, at each stage, only one-dimensional projections of the constellation are available, so the algorithm cannot take full advantage of the distance properties of the two-dimensional constellation. The only complex-domain sphere decoding algorithm was described in [23]. This work considered iterative (turbo) decoding in a MIMO system where linear ST mapping was combined with an outer channel code, and the idea of the sphere detector was used to approximate the log-likelihood ratio in a computationally efficient way. The proposed approach was specific to PSK modulation, or modulations that can be decomposed into PSK modulations. Finally, the most efficient previously proposed algorithm [79] enumerates all possible x_k values that satisfy the current partial constraint, and then sorts them. Since most

candidate values will be eliminated during the decoding process, enumerating and sorting them creates unnecessary computational overhead.

Motivated by these observations, in this work, we propose a general framework for decoding SF block codes constructed from orthogonal and block-orthogonal designs. The framework is formulated in the complex domain, and it can be used with any memoryless modulation method. To achieve flexibility, we also propose a decoding algorithm that does not utilize any information about the used constellation, so it can decode SF codes using any memoryless modulation. Moreover, to maximize computational efficiency, we propose a fast searching algorithm that fully takes advantage of the chosen modulation method and avoids the overhead caused by unnecessary enumeration and searching.

We have divided the description of the algorithm into two stages. The first stage, the preprocessing stage, calculates the zero forcing solution vector, the Cholesky factorization, and produces the needed quantities for the searching stage. The second stage actually performs the search for the decoded channel symbol vector.

4.5.1 The Decoding Algorithm I: Preprocessing Stage

The task of this stage is similar to the computations described in the first part of Section 4.4. The main difference is that now all quantities involved are complex.

First, we calculate the complex Cholesky factorization of the matrix $\mathbf{H}^{\mathcal{H}}\mathbf{H}$, obtaining an $N \times N$ complex, upper triangular matrix $\mathbf{R} = \{R_{k,l}\}$ with real and positive entries along the main diagonal. A possible implementation of the decomposition algorithm will be given later for the purpose of complexity analysis. Then, the zero-forcing solution $\mathbf{z} = \mathbf{H}^{+}\mathbf{y}$ needs to be calculated. Note that this can be

done more efficiently than the straight calculation $\mathbf{z} = (\mathbf{H}^{\mathcal{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathcal{H}}\mathbf{y}$. Calculating the vector $\mathbf{z}$ is equivalent to finding a vector $\mathbf{z}$ such that

$$\mathbf{H}^{\mathcal{H}}\mathbf{H}\mathbf{z} = \mathbf{H}^{\mathcal{H}}\mathbf{y}. \tag{4.16}$$

By defining a vector $\mathbf{b}$ as $\mathbf{b} = \mathbf{H}^{\mathcal{H}}\mathbf{y}$, and using the Cholesky decomposition $\mathbf{H}^{\mathcal{H}}\mathbf{H} = \mathbf{R}^{\mathcal{H}}\mathbf{R}$, (4.16) can be rewritten as

$$\mathbf{R}^{\mathcal{H}}\mathbf{R}\mathbf{z} = \mathbf{b}.$$

As a consequence, the vector $\mathbf{z}$ can be obtained as follows.

1) Calculate $\mathbf{b} = \mathbf{H}^{\mathcal{H}}\mathbf{y}$.

2) Solve the lower triangular system $\mathbf{R}^{\mathcal{H}}\mathbf{w} = \mathbf{b}$ for $\mathbf{w}$.

3) Solve the upper triangular system $\mathbf{R}\mathbf{z} = \mathbf{w}$ for $\mathbf{z}$.

Since steps 2) and 3) only require back substitution ($\mathcal{O}(N^2)$ operations), the matrix inversion or the Gaussian elimination (both $\mathcal{O}(N^3)$ operations) can be avoided, so the complexity is reduced. In case of problems of small size (for example, 2 transmit antennas), the difference is not significant, but for larger systems, considerable complexity reduction can be achieved. The final step is to compute the modified radius

$$r'^2 = r^2 - ||\mathbf{y}||^2 + ||\mathbf{H}\mathbf{z}||^2,$$

and the $N \times N$ complex $\mathbf{Q} = \{Q_{k,l}\}$ matrix according to (4.12).

Complexity Analysis

This subsection compares the computational complexities of the real and complex implementations of the preprocessing stage in terms of the number of real floating

Algorithm component	Number of additions	Number of multiplications	Num. of divisions	Num. of sq. roots
Calculating the $\mathbf{H}^{\mathcal{H}}\mathbf{H}$ matrix	$4MN^2 - 2N^2$	$4MN^2$	0	0
Calculating the Cholesky fact.	$\frac{1}{3}(2N^3+3N^2+N)$	$\frac{1}{3}(2N^3+6N^2+4N)$	N	N
Calculating $\mathbf{b} = \mathbf{H}^{\mathcal{H}}\mathbf{y}$	$4MN - 2N$	$4MN$	0	0
Solving two triang. systems	$4N^2 - 4N$	$4N^2 - 4N$	$4N$	0
Calculating the radius r'^2	$4MN + 2M$	$4MN + 4M$	0	0
Calculating the $\mathbf{Q}$ matrix	0	N	N^2-N	0

Table 4.1: Number of real operations for the complex implementation

```
     L = COMPLEX_CHOLESKY(A)
 1.  for i = 0, 1, ..., N − 1
 2.      for k = i, i + 1, ..., N − 1
 3.          v_k = A_{k,i}
 4.      end for
 5.      for k = 0, 1, ..., i − 1
 6.          for l = i, i + 1, ..., N − 1
 7.              v_l = v_l − L*_{i,k} · L_{l,k}
 8.          end for
 9.      end for
10.      x = 1/√v_i
11.      for l = i, i + 1, ..., N − 1
12.          L_{l,i} = x · v_l
13.      end for
14.  end for
```

Table 4.2: The implemented complex Cholesky decomposition algorithm

point operations. We provide the exact number of real additions (and subtractions), multiplications, divisions and square root operations per code block.

Let us have a take a closer look at the complex implementation first, assuming that the $M \times N$ matrix $\mathbf{H}$ is given. Table 4.1 summarizes the necessary number of real operations for each component of the preprocessing stage. One complex addition was counted as two real additions, and one complex multiplication was counted as four real multiplications and two real additions. Since in the preprocessing stage, complex quantities are divided only by real quantities, one complex division was counted as two real divisions. The square root operation was applied only to real quantities. One squared magnitude operation was counted as two real multiplications and one real addition.

The first step is to calculate the matrix $\mathbf{H}^{\mathcal{H}}\mathbf{H}$. This is followed by the complex Cholesky decomposition algorithm, presented in Table 4.2. The algorithm is a simple extension of the "Gaxpy" version of the real Cholesky decomposition algorithm [83]. The input of the algorithm is an $N \times N$ complex Hermitian, positive definite matrix $\mathbf{A} = \{A_{k,l}\}$, so in our case $\mathbf{A} = \mathbf{H}^{\mathcal{H}}\mathbf{H}$. The output is an $N \times N$ complex, lower triangular matrix $\mathbf{L} = \{L_{k,l}\}$ with real and positive entries along the main diagonal satisfying $\mathbf{L}\mathbf{L}^{\mathcal{H}} = \mathbf{A}$. The upper triangular $\mathbf{R}$ is obtained as $\mathbf{R} = \mathbf{L}^{\mathcal{H}}$. The complex vector $\mathbf{v} = \{v_k\}$ serves as temporary storage. If $\mathbf{A}$ is positive definite, the quantity v_i in line 10. is always positive and real, so the (numerical) rank deficiency of $\mathbf{H}^{\mathcal{H}}\mathbf{H}$ can be checked by checking whether v_i is zero between lines 9. and 10. Having calculated the Cholesky decomposition, we need to calculate the $\mathbf{b}$ vector, and solve two $N \times N$ complex triangular systems to obtain the vector $\mathbf{z}$. Finally, the radius r'^2 and the entries of the $\mathbf{Q}$ matrix need to be calculated.

The computational complexity of the real implementation can be determined

Algorithm component	Number of additions	Number of multiplications	Num. of divisions	Num. of sq. roots
Calculating the $\mathbf{H}_R^T\mathbf{H}_R$ matrix	$8MN^2 - 8N^2$	$8MN^2$	0	0
Calculating the Cholesky fact.	$\frac{4}{3}(N^3 - N)$	$\frac{1}{3}(4N^3+6N^2+2N)$	$2N$	$2N$
Calculating $\mathbf{b}_R = \mathbf{H}_R^T\mathbf{y}_R$	$4MN - 2N$	$4MN$	0	0
Solving two triang. systems	$4N^2 - 2N$	$4N^2 - 2N$	$4N$	0
Calculating the radius r'^2	$4MN + 2M$	$4MN + 4M$	0	0
Calculating the $\mathbf{Q}_R$ matrix	0	$2N$	$2N^2 - N$	0

Table 4.3: Number of real operations for the real implementation

similarly. In this case, all operations are real, but the $\mathbf{H}_R$ matrix has $2M$ rows and $2N$ columns. The algorithm of Table 4.2 can also be used to produce the real Cholesky decomposition of $\mathbf{A} = \mathbf{H}_R^T\mathbf{H}_R$ by replacing N by $2N$, using real operations and ignoring the complex conjugation in line 7. The number of real operations for the real implementation is given in Table 4.3.

Comparing the operation counts in Tables 4.1 and 4.3, it is apparent that the complex implementation of the preprocessing stage has lower computational complexity. Taking into account only the dominant terms MN^2 and N^3, the complex version requires about half of the number of real additions and multiplications. The complex implementation needs $N^2 + 4N$ real divisions per code block, while the real implementation needs $2N^2 + 5N$, approximately twice as many. Finally, the number of square root operations for the complex version is exactly the half of that for the real version. In summary, by implementing the preprocessing stage in the complex domain, for each operation, the operation count can be approximately halved compared to the real implementation.

4.5.2 The Decoding Algorithm II: Searching Stage

Following similar arguments to those described in Section 4.4, the sphere equation can be expressed as

$$||\mathbf{H}(\mathbf{z}-\mathbf{x})||^2 = \sum_{k=0}^{N-1} Q_{k,k} \left| z_k - x_k + \sum_{l=k+1}^{N-1} Q_{k,l}(z_l - x_l) \right|^2, \tag{4.17}$$

where x_k and z_k denote the k-th ($k = 0, 1, \ldots, N-1$) coordinates of the vectors $\mathbf{x}$ and $\mathbf{z}$, respectively. The quantities T_k and α_k, $k = 0, 1, \ldots, N-1$, are now defined as:

$$T_k = \begin{cases} r'^2 & \text{for } k = N-1 \\ T_{k+1} - Q_{k+1,k+1}|\alpha_{k+1} - x_{k+1}|^2 & \text{for } k = N-2, N-3, \ldots, 0, \end{cases} \tag{4.18}$$

and

$$\alpha_k = z_k + \sum_{l=k+1}^{N-1} Q_{k,l}(z_l - x_l) = z_k + \sum_{l=k+1}^{N-1} Q_{k,l}\Delta_l, \tag{4.19}$$

where $\Delta_k = z_k - x_k$ is defined for computational convenience.

The task of the search stage is to go over the coordinates of $\mathbf{x}$ sequentially and enumerate only those x_k values that satisfy the current partial constraint. However, since the objective is to devise a sphere decoding framework that can be used with an arbitrary constellation, the coordinates x_k may take on any complex value. As a consequence, the method described in Section 4.4 cannot be used to find upper and lower bounds on the values of x_k. Moreover, the original sphere decoding algorithm [75] was derived from an algorithm that was designed for finding the shortest vector in a lattice [81]. In our case, the x_k values are not integers, so the set of vectors $\{\mathbf{Hx}\}$ does not form a lattice.

To get around the above problems, we provide an alternative interpretation for the sphere decoding algorithm. We abandon the lattice concept and look at

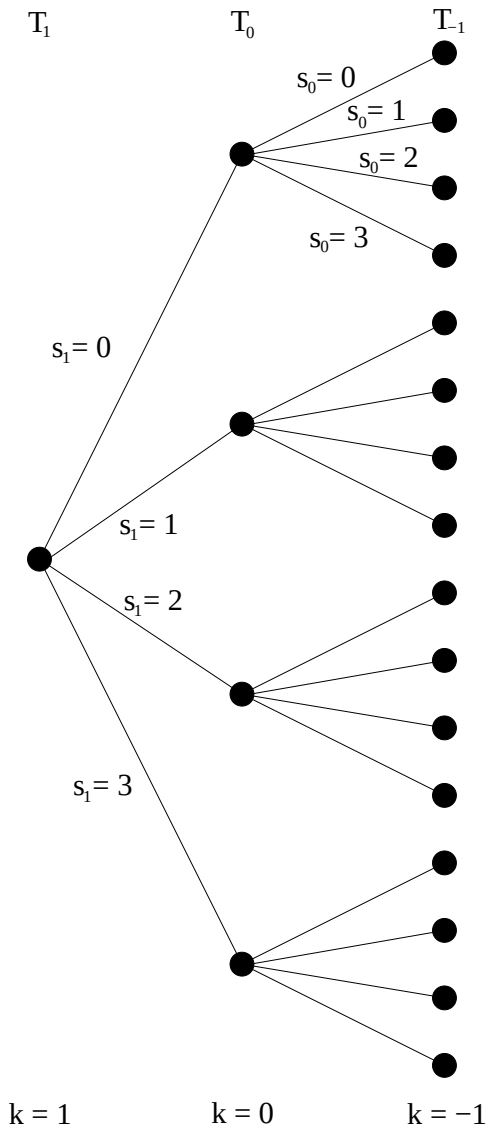

Figure 4.1: Tree representation of the search space

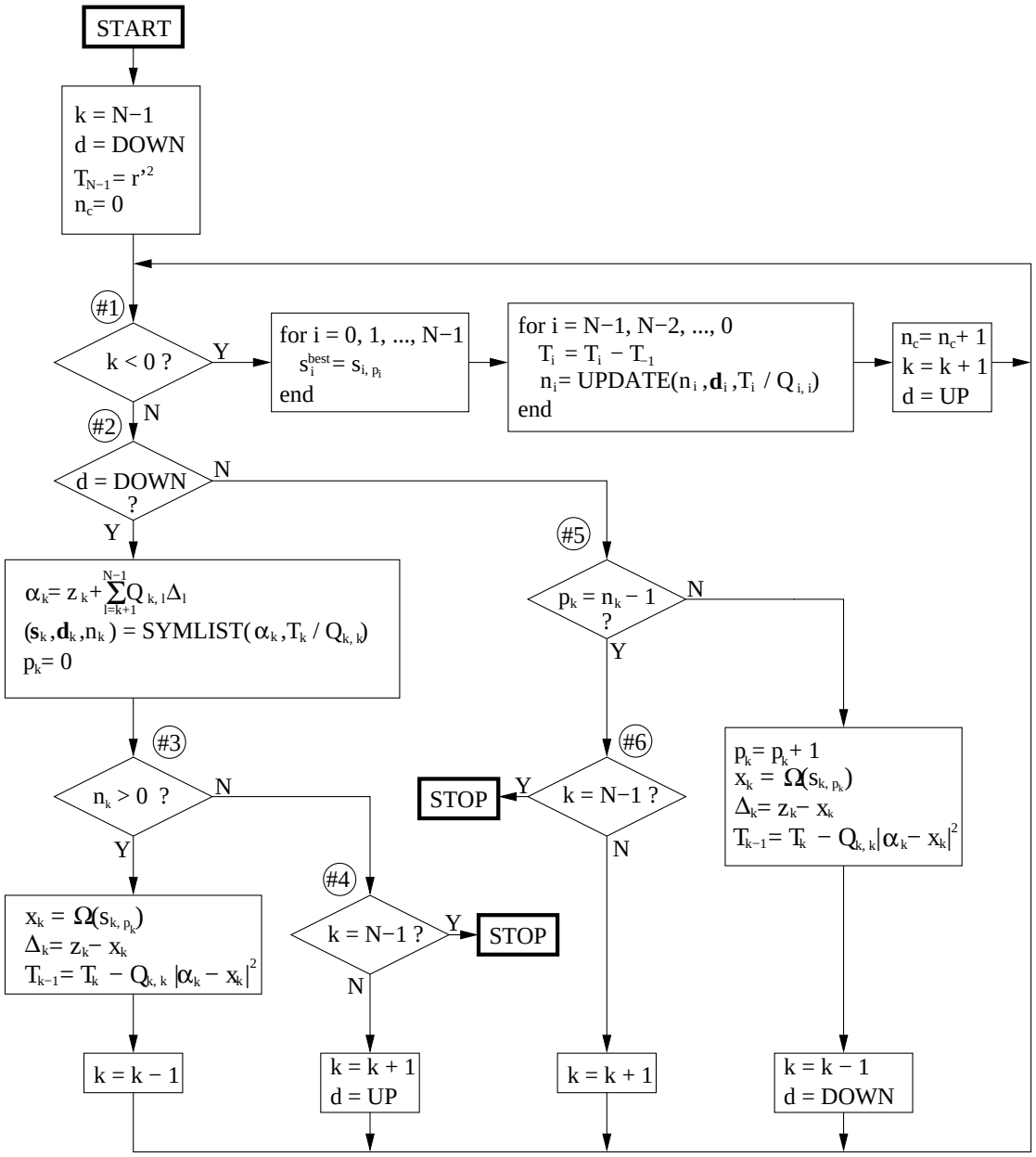

Figure 4.2: The flowchart of the decoding algorithm

sphere decoding as a discrete optimization problem. The objective function to be minimized is $||\mathbf{y} - \mathbf{Hx}||^2$ over all $\mathbf{x} \in \mathcal{X}$ vectors. The search space to be explored can be represented by a tree, as shown in Figure 4.1 in case of the $N = 2$, $B = 4$ example. From the root of the node (level $k = 1$), there are B branches leading to B nodes corresponding to the different values of the source symbol s_1, forming the 0th level ($k = 0$) of the tree. The source symbol values s_1 are mapped to channel symbols according to $x_1 = \Omega(s_1)$. For each value of s_1, there are B possible values of the source symbol s_0, forming the last (-1st) level of the tree. The corresponding channel symbol is $x_0 = \Omega(s_0)$. The tree levels have been indexed in such a way that at level k, the possible values of s_k (and x_k) are enumerated. If we do not take the sphere constraint into account, exploring all the leaf nodes of the tree corresponds to exhaustive search. However, at level k, the partial constraint T_k eliminates some of the branches, so those branches will not have to be explored. Moreover, by finding a valid candidate $\mathbf{x}$ (visiting one of the leaf nodes), may also result in further search space reduction, as the radius of the sphere can be decreased. This suggests that the tree should be explored in a depth-first manner. The dummy constraint T_{-1} represents the "surplus" constraint value after a candidate solution has been found, and its value will be used to decrease the sphere radius. Finally, by exploring the nodes at level k in the order of the increasing metrics $d_k = |\alpha_k - x_k|^2$, we perform a greedy search on the tree. For these reasons, the proposed sphere decoding algorithm can be interpreted as a greedy, constrained, depth-first search algorithm.

The flowchart of the algorithm can be observed in Figure 4.2. The branching points have been numbered for easier explanation. The variable k indexes the levels (the depth) of the tree and the coordinates of the $\mathbf{x}$ vector. The variable

d indicates the direction of the search. If its value is "DOWN", then the node currently being explored is visited for the first time: it was reached from a parent node from above. It the value of d is "UP", the current node has been visited before, and it was reached again from a child node below. The two cases need to be handled differently since the list of possible x_k values is generated only when a node at level k is visited for the first time.

The algorithm starts at the root of the tree, and initializes the necessary variables. We begin the exploration with the last coordinate, x_{N-1}. The variable n_c counts the number of candidate solutions that have been generated (the number of times the tree-search reaches the bottom level). Then, we enter the main loop of the algorithm. At point (#1), the level index k is checked. If it is non-negative, the bottom level of the tree has not been reached yet, so we need to continue the exploration. At point (#2), the algorithm checks whether d="DOWN", i.e. whether the current node is visited for the first time. If so, the current value of α_k is calculated, and then the function SYMLIST generates the list of possible source symbols $s_{k,i}$, whose corresponding channel symbols $x_{k,i} = \Omega(s_{k,i})$ satisfy the normalized partial constraint $|\alpha_k - x_{k,i}|^2 \leq T_k/Q_{k,k}$. The SYMLIST function takes α_k and the normalized partial constraint $T_k/Q_{k,k}$ as inputs and produces 3 outputs. The first output, n_k ($0 \leq n_k \leq B$), is the number of symbols satisfying the current partial constraint, so there will be n_k branches emanating from the current node. The second output is the symbol list $\mathbf{s}_k = [\, s_{k,0}, s_{k,1}, \ldots, s_{k,n_k-1} \,]^T$, and the vector $\mathbf{d}_k = [\, d_{k,0}, d_{k,1}, \ldots, d_{k,n_k-1} \,]^T$ whose coordinates are the metrics $d_{k,i} = |\alpha_k - x_{k,i}|^2$. The symbols in $\mathbf{s}_k$ are ordered according to increasing $d_{k,i}$ values. The last step of this block is to initialize the k-th symbol pointer, p_k, to point to the first element of $\mathbf{s}_k$.

At point (#3), the algorithm checks whether there are any symbols on the list. If there is at least one, we take the first source symbol from the list, calculate the corresponding channel symbol, update the current Δ_k value, determine the partial constraint for the next level, and move one level down on the tree. Since the direction was previously set to 'DOWN", there is no need to change it. If there are no symbols on the list, we reach to point (#4), where we check the current level. If it is the top level ($k = N - 1$), the algorithm terminates without solution. If it is not the top level, we change directions and move back up to the parent node of the current node to continue the search.

If the direction is "UP" at point (#2), the current node has been visited before. In this case, we reach point (#5) to check whether all source symbols (all branches emanating from this node) have been explored. If there are source symbols left on the list, we increment the symbol pointer p_k to point to the next symbol, take the next source symbol from the list, calculate the corresponding channel symbol, update the current Δ_k value, determine the partial constraint for the next level, and move one level down on the tree. Since we change direction, the value of d has to be set to "DOWN". If there are no source symbols left on the list, we get to point (#6) to check whether we are at the top level ($k = N - 1$). If so, the algorithm terminates. If the value of n_c is zero, no solution was found, otherwise the algorithm was able to find at least one $\mathbf{x}$ vector satisfying the sphere constraint. If the algorithm has not reached the top level yet, it moves one level up to the parent node to continue the search. The value of d was previously set to "UP", so we do not need to change it.

Finally, if the value of k becomes negative at point (#1), we have reached the bottom level, so a valid candidate solution $\mathbf{x}$ has been found. Since the radius is

```
     n = UPDATE(n, d, T)
1.   i = 0
2.   while d_i ≤ T & i < n
3.      i = i + 1
4.   end for
5.   return(i)
```

Table 4.4: The implemented UPDATE algorithm

reduced each time a candidate solution is found, the last $\mathbf{x}$ vector is the best solution (i.e. closest to the center) so far, so the corresponding source symbol vector is saved by overwriting the previous solution. Then, all partial constraint values are adjusted such that the last found solution satisfies the constraints with equality (the "surplus" partial constraint is subtracted from each partial constraint). The source symbol lists are also modified by the UPDATE function, which keeps only those source symbols on the list whose corresponding d_k metric values satisfy the new partial constraints. Since the symbols are ordered according to the corresponding d_k values, this can be done simply by changing the value of n_k at each level. The implemented UPDATE function is shown in Table 4.4. It simply goes over the coordinates d_i of the metric vector $\mathbf{d}$ until it encounters one that does not satisfy the constraint T or reaches the end of the list. The symbol "&" here stands for the logical AND operation. After updating the constraints and possibly reducing the search space by shortening the symbol lists, the algorithm increments the number of candidate solutions found (n_c), and moves back up on the tree to the parent node of the current leaf node to continue the search.

Note that the functions SYMLIST and UPDATE have been extracted from the flowchart to keep the presentation simple, but they need not be implemented as functions to reduce the overhead caused by function calls and returns.

Since the algorithm makes decisions based on variables that are functions of

```
     (s,d,n) = SYMLIST(α, T)
 1.  n = 0
 2.  for s = 0, 1, ..., B − 1
 3.     x = Ω(s)
 4.     d = |α − x|^2
 5.     if d > T
 6.        continue
 7.     end if
 8.     i = 0
 9.     while d_i ≤ d & i < n
10.        i = i + 1
11.     end while
12.     n = n + 1
13.     for j = n − 2, n − 3, ..., i
14.        d_{j+1} = d_j
15.        s_{j+1} = s_j
16.     end for
17.     d_i = d
18.     s_i = s
19.  end for
20.  return(s,d,n)
```

Table 4.5: The implemented modulation-independent symbol list generation algorithm

random quantities, it is very hard to calculate its computational complexity accurately. Other researchers could only determine approximate asymptotic results [81] on the order of the number of operations, or provide formulas for the approximate complexity that can only be evaluated numerically [82]. Both works assumed lattice structure, i.e. the coordinates of $\mathbf{x}$ were integers, which is not true in our case. Moreover, asymptotic results on the order of the operations are not appropriate for the comparison of two sphere decoding algorithms, as they may hide any constant factors, and the dimensionality of the problem (the values M and N) remains strictly bounded. Therefore, we have chosen simulation based complexity comparison by counting the number of operations and averaging them over a large number of experiments.

The algorithm of Figure 4.2 is only a general framework that performs a greedy, constrained, depth-first search. The heart of the algorithm is the function SYMLIST, which creates the source symbol list. The efficiency of the sphere decoding algorithm largely depends on the efficiency of this function. Moreover, by using different implementations of the SYMLIST function, the general framework can be tailored to the particular properties of the chosen constellation. In the following subsections, this possibility will be explored further.

4.5.3 Modulation-Independent Search

First, we assume that there is no information available on the used constellation, so we have to develop an algorithm that would work with any memoryless modulation method. In this case, to create the source symbol list, we need to enumerate all possible source symbols, and sort the ones that satisfy the partial constraint, and put them onto the symbol list. The algorithm shown in Table 4.5 does exactly this

using the idea of the insertion sort algorithm [84]. The inputs are the value of α and the partial constraint T, and the outputs are n, the number of symbols on the list, the symbol list $\mathbf{s} = [\, s_0, s_1, \ldots, s_{n-1} \,]^T$, and the corresponding metric list $\mathbf{d} = [\, d_0, d_1, \ldots, d_{n-1} \,]^T$. The algorithm goes over all possible source symbols, calculates the corresponding channel symbols and checks whether the partial constraint T is satisfied. If not, the "continue" instruction jumps to line 19. and the execution continues with the next iteration on s. If the constraint is satisfied, the source symbol s is inserted into the list $\mathbf{s}$ according to the corresponding d metric. Lines 8-11. find the place of the new entry on the list, and the location is the value of i. Then, the number of items on the list gets incremented, and all elements on the list from the i-th position are shifted to accommodate the new symbol (lines 13-16.). Finally, the new symbol and the corresponding metric are inserted into the list.

This algorithm is not expected to perform very well, as it does not take advantage of the geometry of the used constellation. However, even without any information on the applied modulation method, the computational complexity can be significantly reduced compared to the ML decoding algorithm, and this version of the algorithm can be used as a baseline system for the algorithms to be presented in the next subsections.

4.5.4 A Search Method for PSK

The authors of [23] considered the problem of calculating the log-likelihood ratio in an efficient way for iterative decoding. They applied the idea of sphere decoding to limit the number of bit sequences to search over. The problem of finding all signal points of a PSK constellation that lie closer to the point α than a threshold

T was solved in the following way.

Assume that the used modulation method is B-ary PSK, so the modulation mapping is given by

$$x = \Omega(s) = re^{j\phi} = re^{j(\frac{2\pi}{B}s+\delta)}, \quad s \in \{0, 1, \ldots, B-1\}. \tag{4.20}$$

In (4.20), r is the radius of the constellation (usually unity), and δ is the rotation angle of the constellation (usually zero). The average energy of the constellation is $E_{avg} = r^2$. By calculating the magnitude and angle of α as

$$\begin{aligned} r_\alpha &= |\alpha| \\ \phi_\alpha &= \text{arc tan}(\text{Im}\{\alpha\},\text{Re}\{\alpha\}), \end{aligned}$$

where arc tan(.,.) is the four-quadrant arcus tangent function, the constraint on the signal point x can be expressed as [23]

$$|\alpha - x|^2 = r^2 + r_\alpha^2 - 2rr_\alpha \cos(\phi - \phi_\alpha) \leq T,$$

which yields the condition

$$\cos(\phi - \phi_\alpha) \geq \eta = \frac{1}{2rr_\alpha}(r^2 + r_\alpha^2 - T).$$

If $\eta > 1$, there are no signal points satisfying the partial constraint, and if $\eta < -1$, all signal points satisfy the constraint. If $-1 \leq \eta \leq 1$, those signal points lie inside the circle with center point α and radius $\sqrt{T}$ that satisfy

$$|\phi - \phi_\alpha| \leq \text{arc cos}(\eta).$$

Therefore, the search can be restricted as [23]

$$L \leq s \leq U,$$

where the lower bound L and the upper bound U are given by

$$L = \left\lceil \frac{B}{2\pi}(\phi_\alpha - \text{arc cos}(\eta) - \delta) \right\rceil, \text{ and}$$
$$U = \left\lfloor \frac{B}{2\pi}(\phi_\alpha + \text{arc cos}(\eta) - \delta) \right\rfloor. \tag{4.21}$$

Based on the above discussion, a sphere decoding algorithm can be obtained by using the framework of Figure 4.2 and by constructing a SYMLIST function similar to that in Table 4.5:

1) Calculate the bounds L and U according to (4.21).

2) Go over all source symbols s between L and U and insert them into the symbol list one by one according to the metric $|\alpha - x|^2$.

4.5.5 The Fast Search Method

Since the real implementation of the sphere decoding algorithm works with the one-dimensional projections of the two-dimensional constellation, sorting the one-dimensional signal values at two different levels cannot guarantee that the two-dimensional signal values will be sorted according to the correct order. This suggests that only a complex implementation can take full advantage of the geometry of the used constellation.

Moreover, the previously proposed, most efficient sphere decoding approach [79] enumerates all signal points x satisfying the partial constraint at all levels, and then sorts them. However, it was observed through simulations that most of the time, the first candidate solution $\mathbf{x}$ found by the greedy tree search is actually the ML solution, so only one leaf node is explored during the decoding of one SF code block with high probability. This phenomenon is hard to show analytically, as

the value of α_k at level k depends on not only the received signal and the channel state, but also the previous decisions made on $x_{N-1}, x_{N-2}, \ldots, x_{k+1}$. Nonetheless, this means that enumerating and sorting all signal points that satisfy the partial constraints at all levels is redundant because most source symbols on the symbol lists get eliminated via radius reduction and the subtrees corresponding to their values will never be explored.

To further improve the computational efficiency of the sphere decoding algorithm, we propose a fast search approach by making the most probable case more efficient. The proposed fast search algorithm essentially combines the enumeration and sorting stages by performing nearest neighbor search at each tree level. Therefore, the symbol list generation algorithm SYMLIST enumerates only a few nearest signal points x to α and ignores the signal points that are further away. Then, for those few signal points, the algorithm checks whether they satisfy the current partial constraint one by one. As a consequence, we can avoid the unnecessary enumeration and sorting operations. At low SNR, this approach may cause performance degradation, but at higher SNR, it is possible to gain considerable complexity reduction and still achieve the ML decoding performance.

Square QAM Modulation

If the used constellation is a B-ary square QAM, it is most convenient to decompose the complex quantities to real and imaginary parts. However, at each level, the decision on the complex symbols is made taking both the real and imaginary parts into account, as opposed to the real implementation of Section 4.4, where the decisions on the real and imaginary parts of the symbols are made separately at different tree levels.

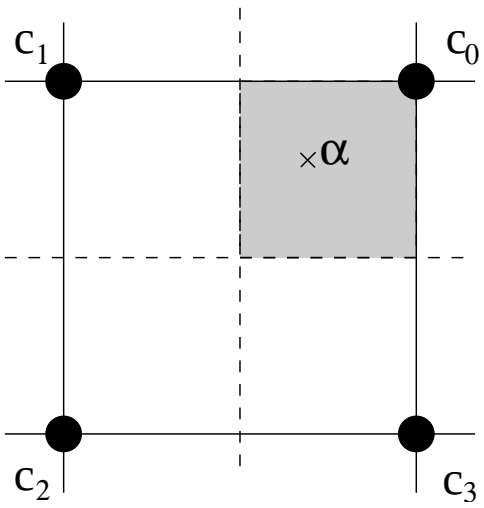

Figure 4.3: The fast QAM search algorithm

The source symbol s is then the concatenation of its real and imaginary parts as: $s = (s^R, s^I)$, $s^R, s^I \in \{0, 1, \ldots, \sqrt{B} - 1\}$, and the complex channel symbol corresponding to this source symbol is

$$x = \Omega(s) = \left(s^R - \frac{\sqrt{B} - 1}{2} \right) + j \left(s^I - \frac{\sqrt{B} - 1}{2} \right).$$

The average energy of the constellation is $E_{avg} = \frac{B-1}{6}$. For the simplicity of the explanation, we neglect the edge effects, and assume that the point α falls in a region where it has four neighboring signal points, as shown in Figure 4.3. The basic idea of the fast search method is to identify these four constellation points efficiently by exploiting the geometrical properties of the QAM constellation. The first step is to express the value of α in the coordinate system of the real and imaginary parts of the source symbols as

$$s_\alpha^R = \mathrm{Re}\{\alpha\} + \frac{\sqrt{B} - 1}{2}, \qquad \text{and} \qquad s_\alpha^I = \mathrm{Im}\{\alpha\} + \frac{\sqrt{B} - 1}{2}.$$

Then, the source symbols corresponding to the four neighboring constellation points can be identified easily. For example, in Figure 4.3, the source symbol

	$(\mathbf{s},\mathbf{d},n) = \text{SYMLIST}(\alpha,\, T)$
1.	$n = 0$
2.	$s_\alpha^R = \text{Re}\{\alpha\} + \frac{\sqrt{B}-1}{2}$
3.	$s_\alpha^I = \text{Im}\{\alpha\} + \frac{\sqrt{B}-1}{2}$
4.	$c_2^R = \lfloor s_\alpha^R \rfloor,\, c_2^I = \lceil s_\alpha^I \rfloor$
5.	$c_0^R = c_2^R + 1,\, c_0^I = c_2^I + 1$
6.	$c_1^R = c_2^R,\, c_1^I = c_0^R$
7.	$c_3^R = c_0^R,\, c_3^I = c_2^I$
8.	$\tilde{s}_\alpha^R = s_\alpha^R - c_2^R$
9.	$\tilde{s}_\alpha^I = s_\alpha^I - c_2^I$
10.	$i = 0$
11.	if $\tilde{s}_\alpha^R > 0.5$
12.	$\quad i = i + 2$
13.	end if
14.	if $\tilde{s}_\alpha^I > 0.5$
15.	$\quad i = i + 1$
16.	end if
17.	for $k = 0, 1, 2, 3$
18.	$\quad l = O_{i,k}$
19.	$\quad x = \Omega(c_l)$
20.	$\quad d = \lvert\alpha - x\rvert^2$
21.	$\quad$ if $d \leq T$
22.	$\quad\quad s_n = c_l$
23.	$\quad\quad d_n = d$
24.	$\quad\quad n = n + 1$
25.	$\quad$ end if
26.	end for
27.	return$(\mathbf{s},\mathbf{d},n)$

Table 4.6: The implemented QAM-specific symbol list generation algorithm

$c_0 = (c_0^R, c_0^I)$ corresponding to the upper right signal point can be obtained as

$$c_0^R = \lceil s_\alpha^R \rceil, \qquad \text{and} \qquad c_0^I = \lceil s_\alpha^I \rceil,$$

which is also the source symbol whose corresponding channel symbol is the closest to α, so assuming that it satisfies the partial constraint, this source symbol should be the first on the symbol list. After determining the 4 source symbols $c_i = (c_i^R, c_i^I)$, $i = 0, 1, 2, 3$, that correspond to the 4 nearest neighbors of α, we need to determine the order in which they should be put on the list. This could be done, for example, by sorting them according to the metric values $|\alpha - \Omega(c_i)|^2$. However, our simulations have shown that the actual order of the symbols is not important as long as the source symbol corresponding to the closest signal point (the greedy solution) is the first on the list. Therefore, we have implemented a lookup table based approximate sorting algorithm. The space between the neighboring QAM constellation points is divided into 4 quadrants, as shown by the dashed lines in Figure 4.3. If, for example, s_α falls into the shadowed quadrant, it is apparent that the closest signal point is the one that corresponds to the source symbol c_0, and the signal point furthest away is the one that corresponds to the source symbol c_2. Based only on this information, the order between c_1 and c_3 cannot be decided, but in this case, we arbitrarily set the order to be c_0, c_1, c_3, c_2.

The source symbol list generating algorithm implementing the nearest neighbor search is shown in Table 4.6. First, the number of elements on the symbol list is initialized, and the value of $s_\alpha = (s_\alpha^R, s_\alpha^I)$ is calculated. Lines 4-7. determine the source symbols corresponding to the 4 nearest constellation points. In lines 8-9., the fractional parts of s_α^R and s_α^I are calculated. These values are used to determine which quadrant the value s_α falls in. Lines 10-16. determine the index of the quadrant, stored in the variable i. Then, we go over all 4 source symbols,

in the order determined by the quadrant index i. The order of the source symbols is stored in a lookup table, given by the matrix $\mathbf{O} = \{O_{i,k}\}$ as

$$\mathbf{O} = \begin{bmatrix} 2 & 3 & 1 & 0 \\ 1 & 2 & 0 & 3 \\ 3 & 0 & 2 & 1 \\ 0 & 1 & 3 & 2 \end{bmatrix}.$$

For each source symbol, the algorithm calculates the corresponding channel symbol and checks whether the partial constraint is satisfied. If so, the symbol and the corresponding metric value are put on the list. Note that the symbols on the list will not be perfectly ordered some of the time. However, the algorithm ensures that the source symbol corresponding to the closest constellation point to α will always be the first.

As can be seen, the algorithm avoids the enumeration of all channel symbols satisfying the partial constraint and avoids the sorting operation altogether. Moreover, many floating point operations have been replaced by integer operations, further reducing the complexity.

PSK Modulation

In case of B-ary PSK, the modulated complex channel symbol is determined by (4.20). The idea is similar to the square QAM case: we identify the source symbols corresponding to the two closest signal points to α without enumerating all that satisfy the partial constraint. As the first step, we express α in the coordinate system of the source symbols, which is an angular coordinate system:

$$s_\alpha = \frac{B}{2\pi}\phi_\alpha - \delta,$$

where ϕ_α is the angle of α, given by

$$\phi_\alpha = \text{arc tan}(\text{Im}\{\alpha\},\text{Re}\{\alpha\}).$$

	$(\mathbf{s},\mathbf{d},n) = \text{SYMLIST}(\alpha,\, T)$
1.	$n = 0$
2.	$\phi_\alpha = \text{arc tan}(\text{Im}\{\alpha\},\text{Re}\{\alpha\})$
3.	$s_\alpha = \frac{B}{2\pi}\phi_\alpha - \delta$
4.	$c_0 = \lfloor s_\alpha \rfloor$
5.	$c_1 = \lceil s_\alpha \rceil$
6.	$\tilde{s}_\alpha = s_\alpha - c_0$
11.	if $\tilde{s}_\alpha > 0.5$
12.	$\quad i = 0$
13.	else
15.	$\quad i = 1$
16.	end if
17.	for $k = 0, 1$
18.	$\quad l = O_{i,k}$
19.	$\quad x = \Omega(c_l)$
20.	$\quad d = \lvert\alpha - x\rvert^2$
21.	$\quad$ if $d \leq T$
22.	$\quad\quad s_n = c_l$
23.	$\quad\quad d_n = d$
24.	$\quad\quad n = n + 1$
25.	$\quad$ end if
26.	end for
27.	return$(\mathbf{s},\mathbf{d},n)$

Table 4.7: The implemented PSK-specific symbol list generation algorithm

Now we can easily determine the two source symbols c_0 and c_1 corresponding to the nearest neighbors of s_α by rounding up and down its value. For easy explanation, the edge effect will not be considered, so c_0 and c_1 are assumed to be in the set $\{0, 1, \ldots, B-1\}$. If they fall outside, we can easily map them back by adding or subtracting B to/from their values. Then, $\tilde{s}_\alpha$, the fractional part of s_α is calculated to determine which neighbor is the closest to α and the source symbol corresponding to the closest signal point is put on the list first, provided that it satisfies the partial constraint. The pseudo-code implementing the PSK-specific symbol list generation algorithm with fast nearest neighbor search is shown in Table 4.7. In case of this algorithm, the matrix $\mathbf{O} = \{O_{i,k}\}$ governing the order of the source symbols is defined as

$$\mathbf{O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

4.6 Simulation Results

To illustrate the performance of the proposed sphere decoding algorithms, we provide some simulation results. The simulated communication system had 2 transmit antennas ($K = 2$) and two receive antennas ($L = 2$). The 2×2 orthogonal design (4.6) was adopted as a SF code, and the equivalent received signal model (4.8) was used for the sphere decoding algorithm with $M = KL = 4$ and $N = n_s = 2$. The OFDM modulation had 128 sub-carriers with an OFDM symbol period of 128μs. The frequency selective MIMO channel was modeled by the COST 207 Typical Urban, 6-ray power delay profile [3]. The channel was assumed spatially uncorrelated and constant over one OFDM symbol period. For all sphere decoding algorithms, the initial radius r was set to 10 to ensure ML performance.

Algorithm component	Number of additions	Number of multiplications	Number of divisions	Number of square roots
Calculating the $\mathbf{H}^{\mathcal{H}}\mathbf{H}$ matrix	72	64	0	0
Calculating the Cholesky decomp.	10	16	2	2
Calculating $\mathbf{b} = \mathbf{H}^{\mathcal{H}}\mathbf{y}$	28	32	0	0
Solving two triang. systems	8	8	8	0
Calculating the radius r'^2	40	48	0	0
Calculating the $\mathbf{Q}$ matrix	0	2	2	0
TOTAL	158	170	12	2

Table 4.8: Number of real operations for the complex implementation, M=4, N=2

Algorithm component	Number of additions	Number of multiplications	Number of divisions	Number of square roots
Calculating the $\mathbf{H}_R^T\mathbf{H}_R$ matrix	96	128	0	0
Calculating the Cholesky decomp.	8	20	4	4
Calculating $\mathbf{b}_R = \mathbf{H}_R^T\mathbf{y}_R$	28	32	0	0
Solving two triang. systems	12	12	8	0
Calculating the radius r'^2	40	48	0	0
Calculating the $\mathbf{Q}_R$ matrix	0	4	6	0
TOTAL	184	244	18	4

Table 4.9: Number of real operations for the real implementation, M=4, N=2

4.6.1 Preprocessing Stage

Since the computational complexity of the preprocessing stage is independent of the applied modulation method, it will be discussed separately from the searching stage. The number of real operations for the complex implementation to decode one code block (two source symbols) is given in Table 4.8. The entries in the table were obtained by substituting $M = 4$ and $N = 2$ into the formulas of Table 4.1. Similarly, Table 4.9 contains the operation counts for the real implementation based on Table 4.3. The operation count values show that considerable complexity reduction can be achieved by implementing the preprocessing stage in the complex domain, even for small M and N values. Especially the number of necessary multiplications and square root operations can be reduced significantly. The total number of floating point operations (FLOPs), including additions, multiplications, divisions and square roots, is 342 for the complex version, while the FLOP count is 450 for the real version. For problems of larger size (larger K and L), even more pronounced complexity reduction is expected based on Tables 4.1 and 4.3. Note that in this particular case, the computational cost of the Cholesky decomposition is negligible. This is due to the small size (2×2) of the matrix to be decomposed.

4.6.2 Searching Stage

We have simulated 3 different systems. The first used 16 QAM modulation, the second used 64 QAM, and the third used 16 PSK-QAM modulation. The latter modulation method will be described in detail later. We have counted the number of different operations and averaged them over a large number ($10^5 - 10^6$) of experiments, and we use the average operation counts to compare the computational complexities of the searching stages of different decoding algorithms. The obtained

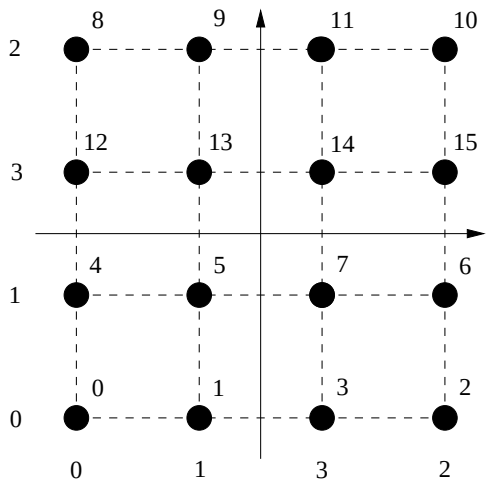

Figure 4.4: The bit mapping for the 16 QAM constellation

numbers may be implementation dependent, but they can be used to estimate the potential advantages and disadvantages of different decoding approaches. In the operation counts, we always compare the number of real operations. The complex operations were counted according to the method described in Section 4.5.1. In addition, one floating point comparison was counted as one addition. Some algorithms also used transcendent functions, such as arcus tangent and arcus cosine; these operations are counted separately. The floating point operation (FLOP) count includes all additions, multiplications, divisions, square roots, comparisons and transcendent function evaluations.

16 QAM

The first set of experiments were conducted with 16 QAM modulation. The used bit mapping is depicted in Figure 4.4. The signal points were labeled with bits according to a one-dimensional Gray mapping along the real and imaginary axes, and the bits were combined such that the imaginary bits occupied the most significant bits and the real bits occupied the least significant bits. The non-natural

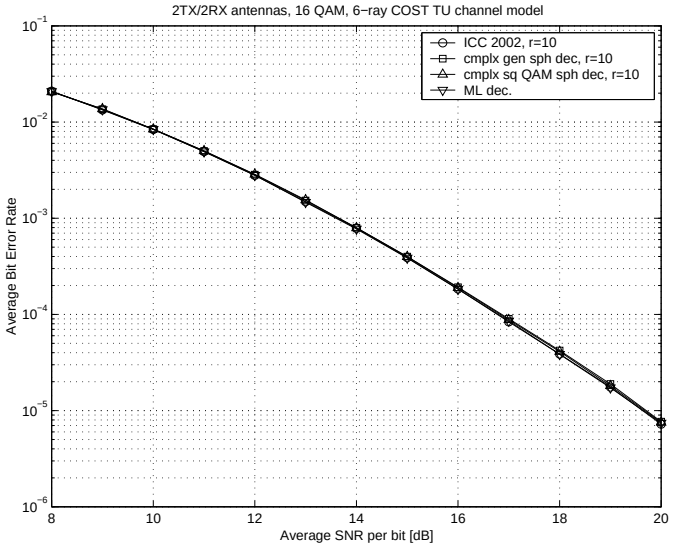

Figure 4.5: Bit error rate of the decoding algorithms with 16 QAM

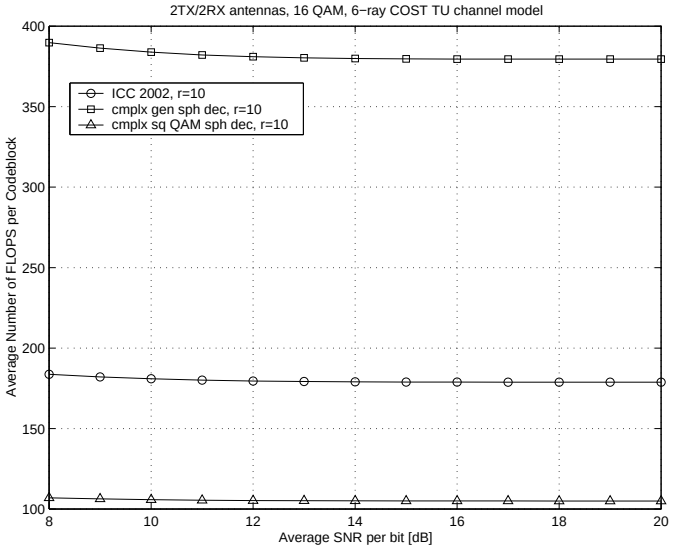

Figure 4.6: Average FLOP counts of the decoding algorithms with 16 QAM

Algorithm	Num. of additions	Num. of multipl.	Num. of divisions	Num. of sq. roots	Num. of FLOPS	n_c
ML decoding	10496	10240	0	0	20736	256
Sphere decoding algorithm of [79]	127.5	36.2	8.1	8.1	179.7	1.00
Sphere decoding, modulation indep.	302.8	74.7	4.0	0	381.6	1.00
Sphere decoding, fast search	75.3	26.1	4.0	0	105.4	1.00

Table 4.10: Number of operations for the 16 QAM case

bit mapping makes the decoding algorithm a little more complicated; nonetheless, the earlier described algorithms can easily be used with minor modification. Note also that the bit mapping operations involve only integer quantities, so they will not affect the floating point operation counts.

We compared four different decoding algorithms: the ML decoding algorithm (performing exhaustive search), the real-domain sphere decoding algorithm described in [79], the complex-domain sphere decoding algorithm with modulation independent symbol list generation method of Section 4.5.3, and the complex-domain sphere decoding algorithm with QAM-specific, fast-search-based symbol list generation method of Section 4.5.5. Figure 4.5 shows the average bit error rate curves as the function of the average signal-to-noise (SNR) ratio. It is apparent from the figure that all sphere decoding approaches have the same performance as the ML decoding in the SNR range of interest. The average number of FLOPs per decoded code block is depicted in Figure 4.6. The ML complexity is left out for the clarity of the figure. Table 4.10 provides a more detailed break-down of the number of operations. Since the variations of the operation counts is minimal at different SNR values, we only present the average operation counts averaged over the SNRs in the simulated range. Here the computational complexity of the ML

decoding is also presented for comparison. The values of the FLOPs were obtained by averaging the sums of the different operation counts. The value n_c (defined in Section 4.5.2 for one code block) is the average number of evaluated candidate solutions, i.e. the average number of times a leaf node of the tree is reached during the search.

Based on the data provided in the Figures 4.5, 4.6 and the Table 4.10, we can make several observations. First, by using a sphere decoder, the computational complexity of the SF decoder can be reduced by orders of magnitude without perceptible performance loss in the meaningful bit error rate range. Second, if we compare table 4.10 with Table 4.8, we can deduce that the computational cost of the sphere decoding algorithms is dominated by the preprocessing stage. This observation is very interesting because it contradicts the observation of [75], where it was found that the searching stage has higher complexity. However, in that case the greedy tree search and the fast nearest neighbor symbol list generation algorithms were not implemented. Third, in case of the sphere decoding algorithms, the average value of n_c is always one if rounded to 2 digits. Its actual value is slightly greater that one. Since the probability of a decoding failure is negligible (during the large number of experiments it never happened), the value of n_c never took on the value 0. This means that with high probability, the first candidate solution found by the greedy search was actually the ML solution, and due to the radius shrinking step, the ML solution excluded all the other candidate solutions from the search. As a consequence, the greedy solution is found optimal (in ML sense) most of the time. Fourth, the modulation independent symbol list generating algorithm did not take advantage of the special properties of the QAM constellation, so its complexity was worse than the 2 other QAM-specific sphere

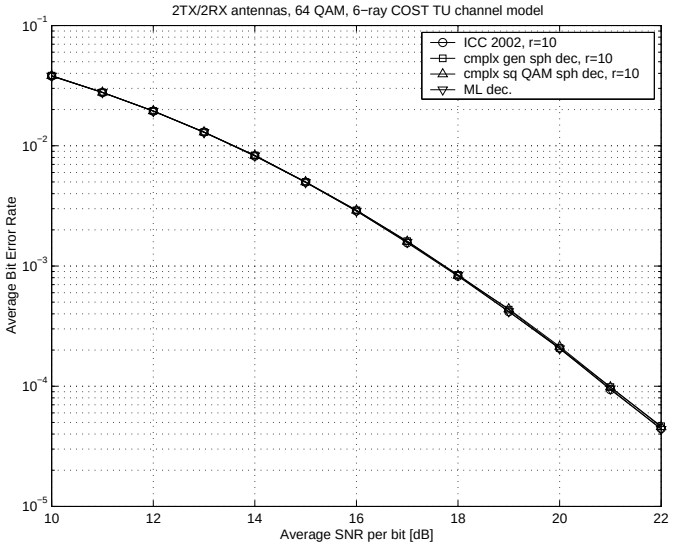

Figure 4.7: Bit error rate of the decoding algorithms with 64 QAM

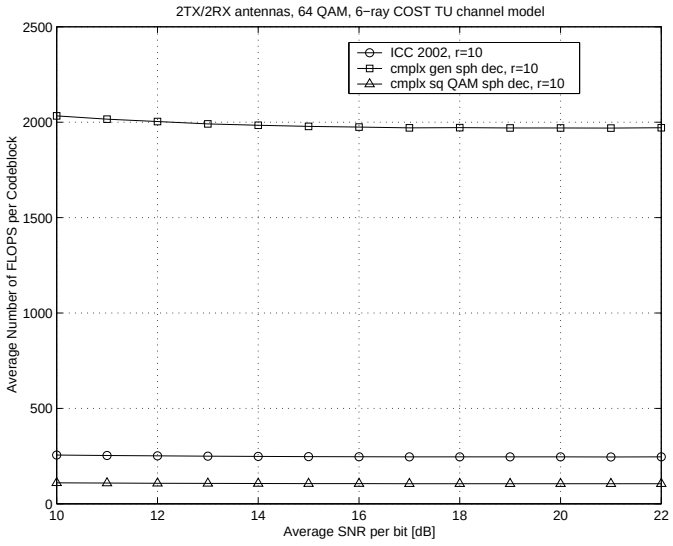

Figure 4.8: Average FLOP counts of the decoding algorithms with 64 QAM

Algorithm	Num. of additions	Num. of multipl.	Num. of divisions	Num. of sq. roots	Num. of FLOPS	n_c
ML decoding	167936	163840	0	0	331776	4096
Sphere decoding algorithm of [79]	183.6	48.5	8.1	8.1	248.3	1.00
Sphere decoding, modulation indep.	1707.6	273.1	4.1	0	1984.8	1.00
Sphere decoding, fast search	76.1	26.3	4.0	0	106.4	1.00

Table 4.11: Number of operations for the 64 QAM case

decoding approaches, as expected. However, we would like to point out that even without any knowledge about the used constellation, significant complexity reduction could be achieved compared to ML decoding. It is also worth mentioning the despite its relatively high FLOP count, the modulation-independent decoder could achieve very low division count and it did not use any square root operations. Finally, by using the proposed complex sphere decoding framework and the fast, QAM-specific nearest neighbor search algorithm to generate the symbol lists, we could get rid of the unnecessary enumerations and sorting operations, and considerable complexity reduction was achieved compared to the algorithm of [79]. We observe about 40-41reduction in the average number of FLOP counts throughout the simulated SNR range. Moreover, the average number of divisions was halved, and the the proposed algorithm did not use square root operations at all.

64 QAM

The performance of the communication system was also simulated with 64 QAM modulation. The bit mapping for the 64 QAM was obtained as a product of two one-dimensional Gray mapping, similarly to the 16 QAM case. The bit error rate curves of the 4 schemes can be observed in Figure 4.7. All algorithms achieve the

ML performance in the simulated SNR range. The average number of FLOPs as a function of the SNR is shown in Figure 4.8, and Table 4.11 contains the detailed operation counts for aech algorithm averaged over the simulated SNR range.

The tendencies observed here are similar to those of the 16 QAM case. The complexity reduction achieved by sphere decoding is much more pronounced, which is due to the increased constellation size. However, comparing Tables 4.11 and 4.10, we observe that the complexities of the algorithm of [79] and the modulation independent decoding approach have increased with the constellation size increase. The authors of [75] provided an asymptotic upper bound on the complexity that was independent of the constellation size, suggesting that the actual complexity does not change with the constellation size either. However, this statement is only true when the sphere radius is small, so it can limit the number of signal point falling inside the sphere. In our case, the algorithm stars with a large radius ($r = 10$) to ensure ML performance, so at the early stages of the algorithm, more signal pints are enumerated and sorted, increasing the computational complexity. However, due to the radius reduction step, the complexity increase is not exponential. In case of the algorithm of [79], this increase is moderate (about 38% increase in the overall FLOP count), but for the modulation independent decoding algorithm, it is more pronounced, as a consequence of enumerating all constellation points at each level of the tree-search. The complexity of the fast QAM-specific nearest neighbor search method did not change with the constellation size increase. This is intuitively expected since the number of nearest neighbor signal points of an arbitrary point in the QAM constellation does not change as its size increases. As a consequence, the achieved complexity reduction of the fast search algorithm is more significant than in the 16 QAM case: the average FLOP count was reduced

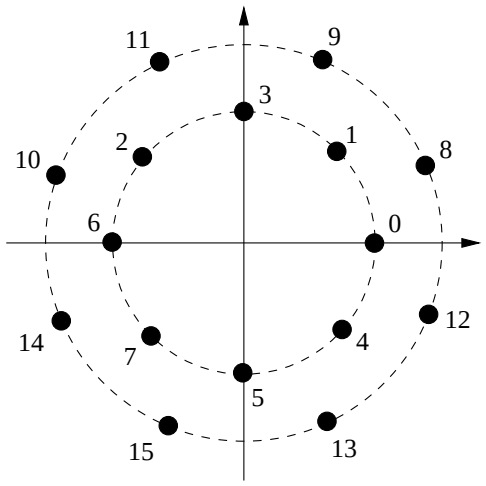

Figure 4.9: The bit mapping for the 16 PSK-QAM constellation

to about 43% of the FLOP count of the algorithm in [79]. The computational complexity of the preprocessing stage still dominates that of the fast nearest neighbor search algorithm.

16 PSK-QAM

The last experiment was conducted with 16 PSK-QAM modulation. The constellation corresponding to this modulation method is shown in Figure 4.9. The 16 PSK-QAM constellation is made up of two PSK constellations. In case of natural bit mapping, the inner PSK signal points can be generated by (4.20) with $r = 1$, $\delta = 0$ and $B = 8$. The outer constellation points are determined by the same equation with $r = \cos(\pi/8) + \sqrt{3}\sin(\pi/8) = 1.5867$, $\delta = \pi/8$ and $B = 8$. We also used a bit mapping that consists of 2 one-dimensional Gray bit mappings for the two PSK constellations with different MSBs. The average energy of the constellation is $E_{avg} = 1.7588$, its minimum distance is $d_{min} = 2\sin(\pi/8) = 0.7654$, and its peak-to-average ratio is $PAPR = 1.43$. (For comparison, the 16 QAM constellation has the following values: $E_{avg} = 2.5$, $d_{min} = 1$ and $PAPR = 1.8$.)

Algorithm	N. of add.	N. of mult.	Num. of divisions	Num. of sq. roots	Num. of tran. op.	Num. of FLOPS	n_c
ML dec.	10496	10240	0	0	0	20736	256
Sphere dec. alg. of [23]	310.8	86.3	8.1	2.0	2.1	409.3	1.00
Sphere dec., mod. indep.	326.0	75.0	4.0	0	0	405.0	1.00
Sphere dec, fast search	73.9	30.2	8.0	0	2.0	114.1	1.00

Table 4.12: Number of operations for the 16 PSK-QAM case

Four different decoding approaches were compared: the ML decoding algorithm, the algorithm constructed using the method of [23] described in section 4.5.4, the proposed sphere decoding algorithm with modulation independent symbol list generation and the proposed sphere decoding algorithm with fast, nearest neighbor search.

The PSK-specific search method of [23] (Section 4.5.4) can easily adapted for the 16 PSK-QAM modulation. When creating the symbol list, we simply go over both the inner and the outer circles, calculate the lower and upper bounds and insert the source symbols on the list. In case of the proposed fast nearest neighbor search, the algorithm described in Section 4.5.5 can also be modified easily. Now there will be 4 nearest neighbor signal points, 2 on the inner circle and 2 on the outer circle, and we will have 2 partial symbol lists that need to be merged. We simply compared the metric values of the first elements on these partial lists, and inserted the partial list with the smaller metric onto the symbol list first. This approach does not guarantee the right ordering all the time, but it was found that as long as the symbol with the smallest metric is the first, the order of the rest of the symbols is not important.

The average bit error rate curves of the above mentioned decoding methods

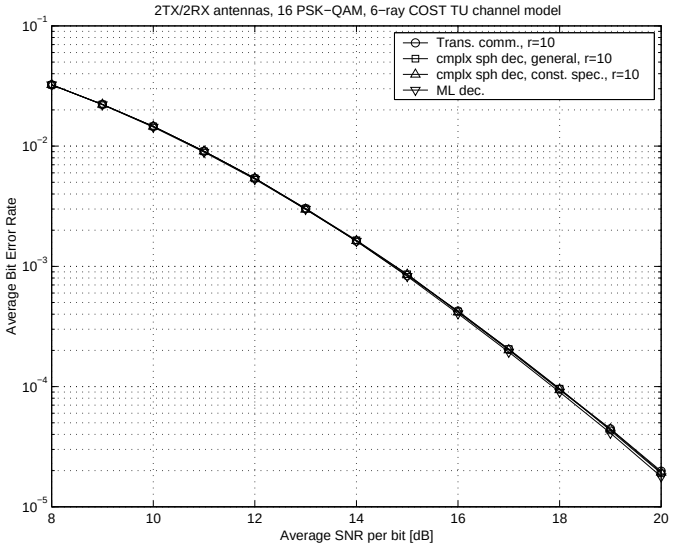

Figure 4.10: Bit error rate of the decoding algorithms with 16 PSK-QAM

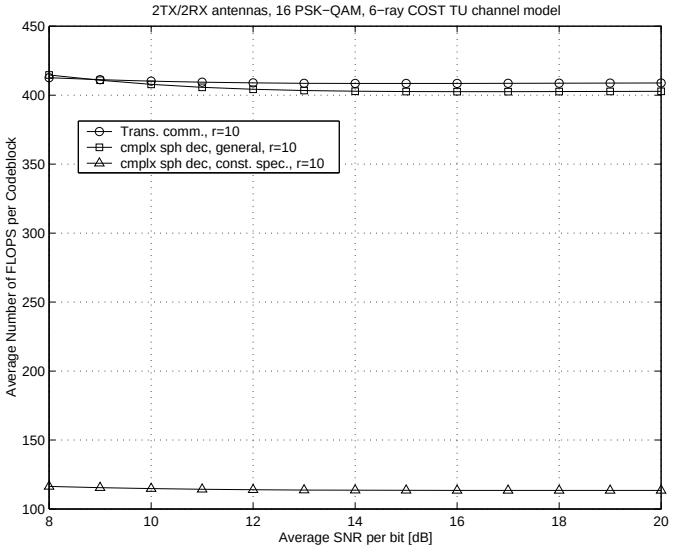

Figure 4.11: Average FLOP counts of the decoding algorithms with 16 PSK-QAM

are depicted in Figure 4.10. It is apparent that all sphere decoding approaches perform as well as the ML decoding algorithm. Moreover, comparing Figures 4.5 and 4.10, we can see that performance of the 16 PSK-QAM modulation is about 1dB worse than the 16 QAM modulation, as expected. The average number of FLOPS per code block as the function of the SNR is shown in Figure 4.11, and the detailed operation counts averaged over the simulated SNR range are given in Table 4.12. The average number of transcendent function evaluations has been added to the table, as some of the decoding algorithms calculate arcus tangent and arcus cosine values.

There are two important observations to be made. First, the constellation independent search performs slightly better than the search method of [23], which takes the special properties of the PSK constellation into account. The reason for this phenomenon is the fact that the latter method is computationally much more expensive than the search method of [79] proposed for QAM modulation. (If we compare Tables 4.12 and 4.10, we can see that the complexity of the modulation independent decoding method is approximately the same for both 16 QAM and 16 PSK-QAM.) However, for constellations of larger size built from PSK constellations, the method of [23] is expected to have lower computational complexity than the modulation independent decoding algorithm. Second, the proposed fast nearest neighbor search algorithm achieved a much more pronounced complexity reduction. The average number of FLOPs was reduced to about 28% of the FLOPs of the decoding method of [23].

4.7 Chapter Summary

We proposed a computationally efficient SF block code decoding algorithm based on the principles of sphere decoding. We formulated the decoding problem in the complex domain and developed a modulation independent decoding framework by interpreting sphere decoding as a greedy, constrained depth-first tree search algorithm. We combined this flexible decoding framework with a modulation independent symbol list generationa algorithm, and two modulation-specific symbol list generation algorithms that perform nearest neighbor signal point search.

By formulating the sphere decoding problem in the complex domain, we could devise a decoding algorithm that can be used with any memoryless modulation method. Moreover, the proposed fast QAM- and PSK-specific decoding algorithms could avoid the unecessary enumerationg and sorting operations, resulting in significant decoding complexity reduction without any perfromance penalty. For example, in case of 64 QAM, the average FLOP count per code block was reduced to about 43%, and in case of 16 PSK-QAM, the average FLOP count per code block was only 28% of that of the best previously known decoding algorithm.

Chapter 5

Conclusions and Future Research

In this thesis, we have examined and explored different aspects of improving the efficiency of multi-antenna wireless communication systems. Specifically, we have proposed two systematic space-time trellis code construction methods for 2 different channel models: the quasi-static and the space-time correlated, flat Rayleigh fading channel models.

For quasi-static channels, we proposed systematic design rules to ensure full diversity by observing and analyzing the group/subgroup structure of the state transitions. We also developed a code construction method that allows for ST code design for both diversity advantage and coding advantage. The flexibility of the method allows us to construct ST codes for an arbitrary number of transmit antennas and any memoryless modulation. Due to the low complexity of the proposed design method, ST codes for a large number of antennas were also constructed. The main conclusion from this work is the following. If the ratio of the maximum and the minimum distances of the chosen constellation is large, our code design method results in codes that substantially outperform the codes that were designed only for diversity advantage.

For the space-time correlated channel model, we derived the performance criteria and developed a systematic ST trellis code design method for an arbitrary number of transmit antennas and any memoryless modulation. The insight we have gained from this work is as follows. First, if the space-time channel is not heavily correlated (i.e. the ST correlation matrix is of full rank), the space-time code design problem for correlated channels can be reduced to the code design problem for fast fading channels. Second, it is possible to devise systematic ST code design methods that result in codes that perform very well compared to codes obtained via computer search. Third, due to the flexibility of the proposed method, we could construct codes for system parameters for which no other ST trellis codes exist in the literature.

Finally, we proposed a computationally efficient SF block code decoding algorithm based on the principles of sphere decoding. We formulated the decoding problem in the complex domain and developed a modulation independent decoding framework by interpreting sphere decoding as a greedy, constrained depth-first tree search algorithm. We combined this flexible decoding framework with a modulation independent symbol list generation algorithm, and two modulation-specific symbol list generation algorithms that perform nearest neighbor signal point search. By avoiding the unnecessary enumerating and sorting operations, our algorithm resulted in significant decoding complexity reduction without any performance penalty.

Based on the research results presented in this dissertation, there are several research directions that can be further investigated:

- **Channel estimation issues:** In this thesis, we have assumed that the perfect channel state information is always available at the receiver side. However, this assumption is not practical, as the channel estimation algorithms can only estimate the MIMO channel with some degree of inaccuracy. Therefore, one possible future direction could be to address various channel estimation issues in connection with ST code design. The first step would be to investigate the effect of channel estimation error on the performance. We can build an analytical framework to model the channel estimation error and derive expressions that characterize how the performance (diversity advantage and coding advantage) degrades due to channel estimation errors. We can also devise code design methods that are robust to channel estimation errors. This would involve deriving new ST code performance criteria taking the channel estimation error explicitly into account by averaging over not only the channel statistics, but also the estimation error statistics. Then, we could identify the important factors that determine the code performance and devise a code construction method that would result in ST codes that perform close to the limits dictated by the channel estimation error. Finally, we can consider the development of joint decoding and channel estimation algorithms. In traditional systems, the channel estimation and data transmission phases are separate. The transmitter first sends a (relatively) long training sequence, and the receiver estimates the channel. Then, the data transmission takes place, assuming that the channel does not change during the transmission of the data block. One possible way to improve the efficiency of the system could be to combine the training and data transmission stages, by observing that based on the decoded data, the receiver can calculate the

transmitted signal, and improve the channel estimate. This approach is also useful for channel tracking when the channel is time varying.

- **ST code design for correlated channels:** During our research described in Chapter 3, we only considered MIMO channels with full-rank space-time correlation matrices. In this case, the effect of code design and and the correlation on the performance could be separated, and channel-independent code design methodology could be developed. However, in most real-world situations, the space-time correlation matrix is rank deficient, so the code design cannot be separated from the channel. This means that the code design will depend on the correlation matrix. As a consequence, it may be worthwhile to explore the following research issues. First, we can derive the ST code performance criteria assuming rank-deficient space-time correlation matrices and devise code design methods taking the correlation matrix into account. Moreover, since the code will depend on the channel correlation, as the mobile moves, the channel changes between the transmitter and the receiver, so the ST code used for communication will also have to be changed. As a consequence, a small-bandwidth feedback channel from the receiver to the transmitter is necessary to inform the transmitter which ST code should be used. This implies the task of designing a system that realizes this feedback communication by answering the following questions: 1.) What kind of information should be sent back? 2.) How many different sets of ST codes should be used and stored in the transmitter and the receiver, and when should they be used? 3.) Is it possible to design robust ST codes, in the sense that one code could guarantee a certain performance over a multitude of channel conditions?

- **Fast SF decoding:** The sphere decoding approach presented in this work assumed that the equivalent channel matrix $\mathbf{H}$ is of full rank. This is a standard assumption in the literature, but in most practical situations, the existence of spatial correlation may make this assumption invalid. This means that some of the diagonal elements of the Cholesky factor $\mathbf{R}$ will be zero, so exhaustive search will have to be performed over the coordinates of the channel symbol vector $\mathbf{x}$ corresponding to those elements. In case of very low-rank channel matrices, the exhaustive search may incur prohibitively large decoding complexity, so the development of new decoding approaches that can avoid the above problem is an important future direction. Moreover, the sphere decoding algorithm has only been proposed by theoretical papers, so it would be interesting to design a fast VLSI architecture that implements it. The Cholesky decomposition algorithm can be replaced by QR decomposition, and the existing VLSI systolic array architectures can be tailored to the needs of this algorithm. The bulk of the work would consist of the implementation of the searching stage by identifying and exploiting the available parallelism and developing multi-processor or pipelined architectures with modular and regular structure.

Appendix A

State Transition Equation I

We will prove (2.5) by induction. Using (2.4), the first state change can be expressed easily:

$$S_1 = B\left(S_0 \text{mod}(RB^{K+p-2})\right) + b_0.$$

Assume that the formula holds for S_{t-1}, formally:

$$S_{t-1} = B^{t-1}(S_0 \bmod (RB^{K+p-t})) + \sum_{m=0}^{t-2} B^{t-2-m} b_m.$$

It only remains to show that the above described relationship also holds for S_t. Using the symbol Q to denote RB^{K+p-2}, S_t can be expressed as:

$$\begin{aligned} S_t = B\left(S_{t-1} \text{mod} Q\right) + b_{t-1} &= B\Big(\Big[B^{t-1}\left(S_0 \text{mod}(RB^{K+p-t})\right) \\ &\quad + \sum_{m=0}^{t-2} B^{t-2-m} b_m\Big] \bmod Q\Big) + b_{t-1}. \end{aligned}$$

By applying the identity

$$\left(\sum_{i=1}^{n} a_i\right) \bmod b = \left(\sum_{i=1}^{n} a_i \bmod b\right) \bmod b,$$

the above expression becomes

$$\begin{aligned} S_t &= B\Big(\Big[(B^{t-1}\left(S_0 \text{mod}(RB^{K+p-t})\right)) \bmod Q \\ &\quad + \sum_{m=0}^{t-2}(B^{t-2-m}b_m) \bmod Q\Big] \bmod Q\Big) + b_{t-1}. \end{aligned}$$

Recognizing that for any $0 \leq n \leq K+p-2$ and $S \in \{0,1,...,N-1\}$

$$(B^n S)\text{mod}(RB^{K+p-2}) = B^n\left(S\text{mod}(RB^{K+p-2-n})\right),$$

the state transition equation can be rewritten as

$$\begin{aligned} S_t &= B\Big(\Big[B^{t-1}\left(\left(S_0 \text{mod}(RB^{K+p-t})\right) \text{mod}(RB^{K+p-t-1})\right) \\ &\quad + \sum_{m=0}^{t-2}(B^{t-2-m}b_m) \bmod Q\Big] \bmod Q\Big) + b_{t-1}. \end{aligned}$$

The next step is to make use of the following simple result: if $b \in \{0,1,\ldots B-1\}$ then for any $0 \leq n < K+p-2$:

$$(B^n b)\text{mod}\left(RB^{K+p-2}\right) = B^n b.$$

This allows for further simplification:

$$\begin{aligned} S_t &= B\Big(\Big[B^{t-1}\left(\left(S_0 \text{mod}(RB^{K+p-t})\right) \text{mod}(RB^{K+p-t-1})\right) \\ &\quad + \sum_{m=0}^{t-2} B^{t-2-m}b_m\Big] \bmod Q\Big) + b_{t-1}. \end{aligned}$$

Now we use the fact that if $S \in \{0,1,...,N-1\}$ and $0 \leq n \leq K+p-1$:

$$\left(S\text{mod}(RB^{K+p-n})\right) \bmod (RB^{K+p-n-1}) = S \bmod (RB^{K+p-n-1}),$$

and we can obtain the following form:

$$S_t = B\left(\left[B^{t-1}\left(S_0 \bmod (RB^{K+p-t-1})\right) + \sum_{m=0}^{t-2} B^{t-2-m}b_m\right] \bmod Q\right) + b_{t-1}.$$

Finally, the identity

$$(B^n x + B^{n-1}b_0 + B^{n-2}b_1 + \ldots + b_{n-1}) \bmod (RB^{K+p-2}) =$$

$$= B^n x + B^{n-1}b_0 + B^{n-2}b_1 + \ldots + b_{n-1}$$

that holds for $b_i \in \{0, 1, ..., B-1\}$, $0 < n \leq K+p-2$, and $x \in \{0, 1, ..., RB^{K+p-2-n}-1\}$ will give us the final step that completes the proof:

$$S_t = B\left(B^{t-1}\left(S_0 \bmod (RB^{K+p-t-1})\right) + \sum_{m=0}^{t-2} B^{t-2-m}b_m\right) + b_{t-1} =$$

$$= B^t\left(S_0 \bmod (RB^{K+p-t-1})\right) + \sum_{m=0}^{t-1} B^{t-1-m}b_m.$$

The closed form expression for S_{K+p} can be derived using similar ideas.

Appendix B

Lower Bound on the Determinant of $\mathbf{\Delta' R' \Delta'}^{\mathcal{H}}$

For simplicity, we assume that $\mathbf{R}$ is p by p, $\mathbf{R'}$ is m by m, and $\mathbf{\Delta'}$ is n by m, with $p \geq m \geq n$. In our case, $p = KLT$, $m = KL\tau$ and $n = L\tau$. Let us denote the positive and real eigenvalues of $\mathbf{R}$ by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. Using the singular value decomposition, $\mathbf{\Delta'}$ can be expressed as $\mathbf{\Delta'} = \mathbf{X}[\mathbf{\Sigma}\ \mathbf{0}]\mathbf{Y}^{\mathcal{H}}$, where $\mathbf{X}$ is an n by n unitary matrix, $\mathbf{Y}$ is an m by m unitary matrix, $\mathbf{\Sigma}$ is an n by n diagonal matrix with the singular values along the diagonal, and $\mathbf{0}$ is an n by $(m-n)$ zero matrix. The matrix $\mathbf{R'}$ admits the spectral decomposition $\mathbf{R'} = \mathbf{U\Lambda' U}^{\mathcal{H}}$, with an m by m unitary matrix $\mathbf{U}$, and a diagonal matrix $\mathbf{\Lambda'} = \mathrm{diag}(\lambda'_1, \lambda'_2, \ldots, \lambda'_m)$. The quantities $\lambda'_1 \geq \lambda'_2 \geq \ldots \geq \lambda'_m$ are the real eigenvalues of $\mathbf{R'}$. We can define $\mathbf{Z}$, the m by m unitary matrix, as $\mathbf{Z} = \mathbf{Y}^{\mathcal{H}}\mathbf{U}$ and partition $\mathbf{Z}$ into an n by m matrix $\mathbf{Z_1}$, and an $(m-n)$ by m matrix $\mathbf{Z_2}$ as $\mathbf{Z} = \begin{bmatrix} \mathbf{Z_1} \\ \mathbf{Z_2} \end{bmatrix}$. The matrix $\mathbf{Q} = \mathbf{Z\Lambda' Z}^{\mathcal{H}}$ will have the same eigenvalues as $\mathbf{R'}$. If $\mathbf{Q}$ is partitioned as $\mathbf{Q} = \begin{bmatrix} \mathbf{Q_{11}} & \mathbf{Q_{12}} \\ \mathbf{Q_{21}} & \mathbf{Q_{22}} \end{bmatrix}$, where $\mathbf{Q_{11}} = \mathbf{Z_1\Lambda' Z_1}^{\mathcal{H}}$ is

an n by n principal submatrix of $\mathbf{Q}$, $\mathbf{\Delta' R' \Delta'}^{\mathcal{H}}$ can be expressed as

$$\mathbf{\Delta' R' \Delta'}^{\mathcal{H}} = \mathbf{X \Sigma Q_{11} \Sigma}^{\mathcal{H}} \mathbf{X}^{\mathcal{H}}. \tag{B.1}$$

Since $\mathbf{\Delta'}$ has full row rank, the matrix $\mathbf{\Sigma}$ has full rank. Using Fisher's inequality [62], it can be easily verified that $\mathbf{Q_{11}}$ also has full rank. Moreover, all matrices on the right hand side of (B.1) are n by n. As a consequence, we have the relationship

$$\det(\mathbf{\Delta' R' \Delta'}^{\mathcal{H}}) = \det(\mathbf{Q_{11}}) \det(\mathbf{\Sigma \Sigma}^{\mathcal{H}}) \det(\mathbf{X X}^{\mathcal{H}}) = \det(\mathbf{Q_{11}}) \det(\mathbf{\Delta' \Delta'}^{\mathcal{H}}). \tag{B.2}$$

To obtain a lower bound on $\det(\mathbf{Q_{11}})$, we use Cauchy's interlacing theorem [73] (also known as the inclusion principle [62]), stated as follows: Let $\mathbf{Q}$ be an m by m Hermitian matrix with real eigenvalues $\lambda'_1 \geq \lambda'_2 \geq \ldots \geq \lambda'_m$. Furthermore, let $\mathbf{Q_{11}}$ be an n by n ($m \geq n$) principal submatrix of $\mathbf{Q}$, with real eigenvalues $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$. Then we have

$$\lambda'_i \geq \mu_i \geq \lambda'_{m-n+i}, \quad \text{for } i = 1, 2, \ldots, n.$$

Moreover, since $\mathbf{R'}$ is a principal submatrix of $\mathbf{R}$, we can apply Cauchy's interlacing theorem to obtain

$$\lambda_i \geq \lambda'_i \geq \lambda_{p-m+i}, \quad \text{for } i = 1, 2, \ldots, m.$$

Therefore, if we form the diagonal matrix $\mathbf{\Lambda}_{min}(n)$ from the n smallest eigenvalues of $\mathbf{R}$ (i.e. $\mathbf{\Lambda}_{min}(n) = \text{diag}(\lambda_{p-n+1}, \lambda_{p-n+2}, \ldots, \lambda_p)$), we obtain the bound

$$\det(\mathbf{Q_{11}}) \geq \det(\mathbf{\Lambda}_{min}(n)). \tag{B.3}$$

Note that (B.3) also shows that $\mathbf{Q_{11}}$ has full rank. Finally, combining (B.2) with (B.3) yields (3.5).

Appendix C

State Transition Equation II

We will prove (3.27) by induction. In Chapter 2, closed form expressions that relate the state sequence $\{S_t\}$ to the starting state S_0 and input source symbol sequence $\{b_t\}$ up to the length of the shortest error event were derived. In case of encoders having N_{min} states, from (2.6), the expression for S_K becomes

$$S_K = \sum_{m=1}^{K-1} B^{K-1-m} b_m, \tag{C.1}$$

showing that the statement is true for $T = 0$. Assume that the formula holds for $T - 1$, i.e.

$$S_{T+K-1} = \sum_{m=T}^{T+K-2} B^{T+K-2-m} b_m. \tag{C.2}$$

The state transition at time t can also be expressed recursively as a function of the previous state S_{t-1}, and the previous source symbol b_{t-1} (Chapter 2, Equation (2.4)) as

$$S_t = B\left(S_{t-1} \bmod (B^{K-2})\right) + b_{t-1}. \tag{C.3}$$

Using (C.3) for $t = T+K$, and combining it with (C.2), we arrive at (3.27), proving that the formula also holds for T.

BIBLIOGRAPHY

[1] R. Steele, *Mobile Radio Communications*, IEEE Press, 1992.

[2] T. Rappaport, *Wireless Communications*, Prentice Hall, 1996.

[3] G. Stuber, *Principles of mobile communication*, Kluwer Academic Publishers, 2001.

[4] S. Saunders, *Antennas and Propagation for Wireless Communication Systems*, John Wiley & Sons Ltd., 1999.

[5] J. Proakis, *Digital communications*, McGraw-Hill, Inc., 1995.

[6] S. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, 1995.

[7] C. Schlegel, *Trellis Coding*, IEEE Press, 1997.

[8] I. Telatar, "Capacity of multi-antenna Gaussian channels", *Technical Report, AT&T Bell Labs*, 1995.

[9] G. Foschini and M. Gans, "On the limits of wireless communication in a fading environment when using multiple antennas", *Wireless Personal Communication*, March 1998.

[10] J. Winters, "On the capacity of radio communication systems with diversity in a Rayleigh fading environment", *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 5, pp. 871-878, June 1987.

[11] J. Winters, J. Salz and R. Gitlin, "The impact of antenna diversity on the capacity of wireless communication systems", *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 1740-1750, February/March/April 1994.

[12] T. Marzetta and B. Hochwald, "Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading", *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 139-157, January 1999.

[13] J. Foschini "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas", *Bell Labs Technical Journal*, pp. 41-59, Autumn 1996.

[14] P. Wolniansky, G. Foschini, G. Golden and R. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel", *Proc. of URSI International Symposium on Signals Systems and Electronics*, pp. 295-300, 1998.

[15] G. Foschini, G. Golden, R. Valenzuela and P. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 11, pp. 1841-1852, Nov. 1999.

[16] G. Golden, G. Foschini, R. Valenzuela and P. Wolniansky , "Detection algorithm and initial laboratory results using V-BLAST space-time commu-

nication architecture", *Electronics Letters*, vol. 35, no. 1, pp. 14-16, Jan. 1999.

[17] F. Farrokhi, A. Lozano, G. Foschini and R. Valenzuela, "Spectral efficiency of FDMA/TDMA wireless systems with transmit and receive antenna arrays", *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 591-599, Oct. 2002.

[18] M. Sellathurai and S. Haykin, "Further results on the diagonal-layered space-time architecture", *Proc. of IEEE VTC*, vol. 3, pp. 1958- 1962, 2001.

[19] N. Prasad and M. Varanasi, "Optimizing the performance of D-BLAST lattice codes for MIMO fading channels", *Proc. of IEEE International Conference on Personal Wireless Communications*, pp. 61-65, 2002.

[20] S. Catreux, P. Driessen and L. Greenstein, "Data throughputs using multiple-input multiple-output (MIMO) techniques in a noise-limited cellular environment", *IEEE Transactions on Wireless Communications*, vol. 1, no. 2, pp. 226-234, Apr. 2002.

[21] S. Chung, A. Lozano, H. Huang, A. Sutivong and J. Cioffi, "Approaching the MIMO capacity with V-BLAST: theory and practice", submitted to *EURASIP Journal on Applied Signal Processing*.

[22] M. Sellathurai and S. Haykin, "Turbo-BLAST for wireless communications: theory and experiments", *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2538 -2546, Oct. 2002.

[23] B. Hochwald and S. ten Brink, "Achieving Near-Capacity on a Multiple-Antenna Channel", *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 389-399, March 2003.

[24] B. Hochwald and T. Marzetta , "Unitary space-time modulation for multiple-antenna communications in Rayleigh flat fading", *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 543-564, March 2000.

[25] B. Hochwald, T. Marzetta, T. Richardson, W. Sweldens and R. Urbanke, "Systematic design of unitary space-time constellations", *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 1962-1973, Sept. 2000.

[26] B. Hughes, "Differential space-time modulation", *IEEE Transactions on Information Theory*, vol. 46, no. 7, pp. 2567-2578, Nov. 2000.

[27] B. Hochwald and W. Sweldens, "Differential unitary space-time modulation", *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2041-2052, Dec. 2000.

[28] T. Josefiak, "Realization of Hurwitz-Radon matrices", *Queen's Papers in Pure and Applied Mathematics*, no. 36, pp.346-351, 1976.

[29] W. Wolfe, "Amicable orthogonal designs - existence", *Canad. J. Math.*, vol.28, pp.1006-1020, 1976.

[30] A. V. Geramita and J. M. Geramita, "Complex orthogonal designs", *Journal of Combinatorial Theory*, Series A (25), pp.211-225, 1978.

[31] A. V. Geramita and J. Seberry, *Orthogonal Designs, Quadratic Forms and Hadamard Matrices, Lecture Notes in Pure and Applied Mathematics,* vol.43, New York and Basel: Marcel Dekker, 1979.

[32] S. Alamouti, "A simple transmit diversity technique for wireless communications", *IEEE JSAC*, vol. 16, no. 8, pp. 1451-1458, 1998.

[33] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs", *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456-1467, 1999.

[34] G. Ganesan and P. Stoica, "Space-time diversity using orthogonal and amicable orthogonal designs", *Proceedings of IEEE ICASSP*, vol. 5, pp. 2561-2564, 2000.

[35] G. Ganesan and P. Stoica, "Space-time block codes: A maximum SNR approach", *IEEE Trans. Inform. Theory*, vol. 47, pp.1650-1656, May 2001.

[36] O. Tirkkonen and A. Hottinen, "Square-matrix embeddable space-time block codes for complex signal constellations", *IEEE Trans. Inform. Theory*, vol. 48, no. 2, pp.384-395, Feb. 2002.

[37] W. Su and X.-G. Xia, "On space-time block codes from complex orthogonal designs", *Wireless Personal Communications (Kluwer Academic Publishers)*, vol. 25, no. 1, pp. 1-26, April 2003.

[38] H. Jafarkhani, "A quasi-orthogonal space-time block code", *IEEE Trans. Commun.*, vol. 49, no. 1, pp.1-4, 2001.

[39] O. Tirkkonen, A. Boariu and A. Hottinen, "Minimal non-orthogonality rate 1 space-time block code for 3+ Tx antennas", *IEEE 6th Int. Symp. on Spread-Spectrum Tech. & Appl. (ISSSTA 2000)*, pp.429-432, September 2000.

[40] N. Sharma and C. B. Papadias, "Improved quasi-orthogonal codes through constellation rotation", *IEEE Trans. Commun.*, vol. 51, no. 3, pp.332-335, March 2003; also in *Proc. ICASSP'02*, Orlando, FL, May 2002.

[41] W. Su and X.-G. Xia, "Quasi-orthogonal space-time block codes with full diversity", *Proc. IEEE GLOBECOM'02*, vol. 2, pp.1098-1102, 2002.

[42] W. Su and X.-G. Xia, "Space-time block codes with full diversity from quasi-orthogonal designs", *IEEE Transactions on Information Theory*, January 2002 (in revision).

[43] Z. Wang and G. Giannakis, "Wireless multicarrier communications - where Fourier meets Shannon", *IEEE Signal Processing Magazine*, pp. 29-48, May 2000.

[44] T. Starr, J. Cioffi and P. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice Hall, 1999.

[45] G. Raleigh and J. Cioffi, "Spatio-temporal coding for wireless communication", *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 357-366, March 1998.

[46] D. Agrawal, V. Tarokh, A. Naguib and N. Seshadri, "Space-time coded OFDM for high data-rate wireless communication over wideband channels", *Proc. IEEE VTC*, vol. 3, pp. 2232-2236, 1998.

[47] K. Lee and D. Williams, "A Space-frequency transmitter diversity technique for OFDM systems", *Proc. IEEE GLOBECOM*, vol. 3, pp. 1473-1477, 2000.

[48] R. Blum, Y. Li, J. Winters and Q. Yan, "Improved space-time coding for MIMO-OFDM wireless communications", *IEEE Transactions on Communications*, vol. 49, no. 11, pp. 1873-1878, Nov. 2001.

[49] Y. Gong and K. B. Letaief, "An efficient space-frequency coded wideband OFDM system for wireless communications", *Proc. IEEE ICC*, vol. 1, pp. 475-479, 2002.

[50] Z. Hong and B. Hughes, "Robust space-time codes for broadband OFDM systems", *Proc. IEEE WCNC*, vol. 1, pp. 105-108, 2002.

[51] H. Bölcskei and A. J. Paulraj, "Space-frequency coded broadband OFDM systems", *Proc. IEEE WCNC*, pp.1-6, Sept. 2000.

[52] H. Bölcskei and A. J. Paulraj, "Space-frequency codes for broadband fading channels", *Proc. ISIT'2001*, p.219, Washington DC, June 24-29, 2001.

[53] B. Lu and X. Wang, "Space-time code design in OFDM systems", *Proc. IEEE GLOBECOM*, pp.1000-1004, Nov. 2000.

[54] Y. Gong and K. B. Letaief, "Space-frequency-time coded OFDM for broadband wireless communications", *Proc. IEEE GLOBECOM*, San Antonio, USA, Nov. 2001.

[55] Z. Liu, Y. Xin and G. Giannakis, "Space-time-frequency coded OFDM over frequency selective fading channels", *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2465-2476, Oct. 2002.

[56] W. Su, Z. Safar, M. Olfat and K. J. R. Liu, "Full-Diversity Space-Frequency Codes for MIMO-OFDM Systems", *IEEE International Symposium on Information Theory*, June 2003.

[57] W. Su, Z. Safar, M. Olfat and K. J. R. Liu, "Obtaining Full-Diversity Space-Frequency Codes from Space-Time Codes via Mapping", to appear, *IEEE Transactions on Signal Processing*, Nov. 2003.

[58] J. Guey, M. Fitz, M. Bell and W. Kuo, "Signal design for transmitter diversity wireless communication systems over Rayleigh fading channels", *IEEE Vehicular Technology Conference*, pp. 136-140, 1996.

[59] V. Tarokh, N. Seshadri, A. Calderbank, "Space-time codes for high data rate wireless communication : performance criterion and code construction", *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744-765, March 1998.

[60] J. Grimm and M. Fitz and J. Krogmeier, "Further results on space-time coding for Rayleigh fading", *Proceedings of the 36th Allerton Conference on Communications, Control and Computing*, pp. 391-400, 1998.

[61] A. R. Hammons and H. El Gamal, "On the theory of space-time codes for PSK modulation", *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 524-542, March 2000.

[62] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[63] Q. Yan and R. Blum, "Optimum space-time convolutional codes", *Wireless Communications and Networking Conference*, vol. 3, pp. 1351-1355, 2000.

[64] S. Baro, G. Bauch and A. Hansmann, "Improved codes for space-time trellis coded modulation", *IEEE Communications Letters*, vol. 4, no. 1, pp. 20-22, January 2000.

[65] Z. Safar and K.J.R. Liu,"Variable rate space-time trellis codes", *IEEE Vehicular Technology Conference (VTC)*, vol. 1, pp. 406-409, Sept. 2002.

[66] S. Zummo and S. Al-Semari, "Space-time coded QPSK for rapid fading channels", *Proceedings of PIMRC*, vol. 1, pp. 504-508, 2000.

[67] W. Firmanto, B. Vucetic and J. Yuan, "Space-time TCM with improved performance on fast fading channels", *IEEE Communications Letters*, vol. 5, no. 4, pp. 154-156, April 2001.

[68] H. Bolcskei and A. Paulraj, "Performance of space-time codes in the presence of spatial fading correlation", *Proceedings of Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 687-693, 2000.

[69] M. Damen, A. Abdi and M. Kaveh, "On the effect of correlated fading on several space-time coding and detection schemes", *Proceedings of VTC (Fall)*, vol. 1, pp. 13-16, 2001.

[70] S. Siwamogsatham, M. Fitz, and J. Grimm, "A new view of performance analysis of transmit diversity schemes in correlated Rayleigh fading", *IEEE Transactions on Information Theory*, vol. 48, no. 4, pp. 950-956, Apr. 2002.

[71] S. Siwamogsatham and M. Fitz, "Robust space-time codes for correlated Rayleigh fading channels", in *Proc. 38th Annual Allerton Conf. Communication, Control, and Computing*, Oct. 2000. Also in *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2408-2416, Oct. 2002.

[72] H. El Gamal, "On the robustness of space-time coding", *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2417-2428, Oct. 2002.

[73] G. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.

[74] Z. Safar and K. J. R. Liu, "Space-time correlation of MIMO flat Rayleigh fading channels", *Proceedings of EUSIPCO*, vol. 3, pp. 323-326, 2002.

[75] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels", *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639-1642, 1997.

[76] M. Damen, A. Chkeif and J. Belfiore, "Lattice code decoder for space-time codes", *IEEE Communication Letters*, vol. 4, no. 5, pp. 161-163, 2000.

[77] M. Damen, K. Abed-Meraim and J. Belfiore, "Generalized lattice decoder for asymmetrical space-time communication architecture", *Proc. of IEEE ICASSP*, vol. 5, pp. 2581-2584, 2000.

[78] M. Damen, K. Abed-Meraim and J. Belfiore, "Generalised sphere decoder for asymmetrical space-time communication architecture", *Electronics Letters*, vol. 36, no. 2, pp. 166-167, Jan 2000.

[79] A. Chan and I. Lee, "A new reduced complexity sphere decoder for multiple antenna systems", *Proc. of IEEE ICC*, vol. 1, pp. 460-464, 2002.

[80] H. Vikalo and B. Hassibi, "Maximum likelihood sequence detection of multiple antenna systems over dispersive channels via sphere decoding", *EURASIP Journal on Applied Signal Processing, Special Issue on Space-Time Coding and Its Applications - Part II*, vol. 2002, no. 5, pp. 525-531, 2002.

[81] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis", *Mathematical Computing*, vol. 44, pp. 463-471, 1985.

[82] B. Hassibi and H. Vikalo, "On the expected complexity of integer least-squares problems", *IEEE ICASSP*, vol. 2, pp. 1497-1500, 2002.

[83] G. Golub and C. Van Loan, *Matrix computations*, Johns Hopkins University Press, 1996.

[84] T. Cormen, C. Leiserson and R. Rivest, *Introduction to algorithms*, MIT Press, 1998.